

שפת C

תרגול אל השיא

לספר זה לא מצורף תקליטור. יש להתעלם
מעמודים 15 – 26 ומכל דבר לגבי התקליטור

את הקבצים של התוכניות תמצא באתר הוד-עמי

www.hod-ami.co.il

בקטגוריה "קבצי תרגול לספרים" במספר 59300

עורך ראשי: **יצחק עמיהוד**

עריכה לשונית ועיצוב: **שרה עמיהוד**

עיצוב עטיפה: **ישראל מצגר**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי עשתה כמיטב יכולתה למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאמה.

הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך כל אחריות שהיא.

המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי אינה אחראית כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהתקליטור שמצורף לו.

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.

☐ טלפון: 09-9564716

☐ פקס: 09-9571582

☐ דואר אלקטרוני: info@hod-ami.co.il

☐ אתר באינטרנט: www.hod-ami.co.il

שפת C

תרגול אל השיא

זיו איילון



C Language - Exercises to the top

By Ziv Ayalon

© כל הזכויות שמורות

הוצאת הוד-עמי

לספרי מחשבים בע"מ

ת.ד. 6108 הרצליה 46160

טלפון: 09-9564716 פקס: 09-9571582

www.hod-ami.co.il

info@hod-ami.co.il

אין להעתיק או לשדר בכל אמצעי שהוא ספר זה או קטעים ממנו בשום צורה ובשום אמצעי אלקטרוני או מכני, לרבות צילום והקלטה, אמצעי אחסון והפצת מידע, ללא אישור בכתב מאת ההוצאה, אלא לשם ציטוט קטעים קצרים בציון שם המקור.

הודפס בישראל 2000

All Rights Reserved

HOD-AMI Ltd.

P.O.B. 6108, Herzliya

ISRAEL, 2000

מסת"ב 965-361-269-7 ISBN

תוכן עניינים מקוצר

11מבוא
15התקליטור המצורף והתוכנה TCLite
27פרק 1: אלגוריתמיקה
47פרק 2: תרשימי זרימה
65פרק 3: כלי הפיתוח של השפה
71פרק 4: אבני היסוד של השפה
81פרק 5: משתנים
101פרק 6: משפטי תנאי
135פרק 7: לולאות
183פרק 8: פונקציות
205פרק 9: מערכים
245פרק 10: מחרוזות

תוכן העניינים

11	מבוא
11	הצורך בתרגול.....
11	ייחוד הספר.....
12	למי מיועד הספר.....
12	ספר הלימוד "שפת C אל השיא".....
12	דרישות.....
13	ארגון הספר.....
13	כיצד להשתמש בספר.....

~~15~~ ~~התקליטור המצורף והתוכנה TCLite~~

15	מה בתקליטור.....
16	התיקיות הרלוונטיות לספר.....
16	קוד המקור.....
16	העתקת קבצי קוד המקור לדיסק הקשיח.....
17	תוכנת TCLite.....
17	התקנת התוכנה.....
18	הפעלת התוכנה.....
19	הרצת תוכניות.....
20	עקרונות העבודה.....
25	חלון העריכה.....
25	הרצת תוכנית דרך שורת DOS.....
26	כתיבת עברית בתוכנית.....
26	מה עוד בתקליטור?.....

27 **פרק 1: אלגוריתמיקה**

27	שאלות.....
27	מושגי תכנות בסיסיים.....
27	האלגוריתם.....
27	כללים בתכנון האלגוריתם.....
28	אלגוריתמים המיועדים לאדם.....
30	אלגוריתמים פשוטים.....
30	הוראת הקלט.....
30	חוקים במתן שמות למשתנים.....
30	הוראת הפלט / הוראת ההשמה.....

31תנאים
31ביצוע מותנה באלגוריתמים המיועדים למחשב
32התנאי המורכב וגם / התנאי המורכב או
32הוראות תנאי מקוננות
32לולאות
32בצע ... פעמים
33עבור ... מ-... עד ... בצע
33כל עוד ... בצע
34בצע ... עד ש
34אלגוריתמים ומערכים
35תשובות

47 פרק 2: תרשימי זרימה

47שאלות
47תרשימי זרימה פשוטים
47תרשימי זרימה המיועדים לאדם
48ייצוג פעולות אלגוריתמיות
48פעולת הקלט
49פעולת הפלט
49פעולת ההשמה
50תרשימי זרימה המיועדים למחשב
50תרשימי זרימה בעלי תנאים
51הוראות תנאי מקוננות
52תרשימי זרימה בעלי לולאות
54תשובות

65 פרק 3: כלי הפיתוח של השפה

65שאלות
65סוגי קבצים
66יצירת הקבצים
66יצירת קובץ ההרצה
68עבודה עם המהדר
68הודעות שגיהא / הודעות אזהרה
68התנסות עם הודעות המהדר
69תשובות

71	פרק 4: אבני היסוד של השפה
71	שאלות
71	תוכנית שלד
71	תווים לבנים
73	פלט בסיסי
74	ירידה לשורה חדשה
74	הערות
75	תווי בקרה
76	מרכיבי השפה
77	תשובות

81	פרק 5: משתנים
81	שאלות
81	מהו משתנה?
82	הצבת ערכים
82	סוגי משתנים
82	המשתנה השלם / המשתנה העשרוני
83	דיוק בערכים שלמים ועשרוניים
83	הגדרת משתנים
84	פלט ערכי משתנים
85	קליטת ערכים מהמשתמש
86	המשתנה התווי
86	פלט תווים / קליטת תווים
87	הפונקציה getch / הפונקציה printf
87	פלט מורכב באמצעות printf
88	אתחול משתנים
88	אופרטורים מקוצרים
88	אופרטורי השמה מקוצרים
88	אופרטורי העלאה והפחתה עצמית
89	ביטויים מורכבים
89	ערך מוחזר מאופרטורי העלאה והפחתה עצמית
89	הערך המוחזר מאופרטורי השמה
90	סוגי משתנים נוספים
91	תשובות

פרק 6: משפטי תנאי 101

101	שאלות
101	ביטויי יחס
102	משפט התנאי if
103	משפט התנאי if-else
104	קינון משפטי if
105	קינון משפטי if-else
107	המשפט switch
111	גלישה
112	שורת ה-default
114	אופרטורים לוגיים
114	אופרטור וגם הלוגי
115	אופרטור או הלוגי
115	אופרטור השלילה הלוגי
115	שילוב אופרטורים לוגיים
116	ערכי ביטויי יחס
118	תשובות

פרק 7: לולאות 135

135	שאלות
135	הלולאה while
137	הלולאה do-while
141	הלולאה for
144	אופרטור הפסיק
146	שילוב לולאות ומשפטי תנאי
147	לולאות מקוננות
150	לולאות אינסופיות
151	משפטי בקרה נוספים
151	המשפט goto
154	המשפט break
157	המשפט continue
160	תשובות

פרק 8: פונקציות 183

183	שאלות
183	הגדרת הפונקציה
184	פונקציות פשוטות
185	סדר ביצוע פונקציות והחזרה מהן
188	פונקציות שמקבלות פרמטרים
189	פונקציות שמחזירות ערכים
190	שימושים נוספים במשפט return

192	הצהרת הפונקציה
193	תשובות

פרק 9: מערכים 205

205	שאלות
205	מערכים חד-מימדיים
209	מערכים דו-מימדיים
214	אתחול מערכים
214	אתחול מערך חד-מימדי
215	אתחול מערך דו-מימדי
217	מערכים רב-מימדיים
219	אתחול מערך רב-מימדי
220	שליחת מערכים לפונקציות
220	שליחת מערך חד-מימדי
221	שליחת מערך דו-מימדי
222	תשובות

פרק 10: מחרוזות 245

245	שאלות
245	הקבוע המחרוזתי
246	אתחול מחרוזות
247	המחרוזות והמשתמש
247	פלט מחרוזות / קליטת מחרוזות / הפונקציה gets
251	פונקציות ספרייה לטיפול במחרוזות
251	מציאת אורך מחרוזת
251	העתקת מחרוזות
252	הוספת מחרוזות
253	השוואת מחרוזות
254	מערכי מחרוזות
254	שליחת מחרוזות לפונקציות
254	שליחת מחרוזת
255	שליחת מערך מחרוזות
256	תשובות

הצורך בתרגול

לימוד תיאורטי של שפת תכנות, אינטנסיבי ככל שיהיה, חסר ערך ללא תרגול רב. ולמרות זאת, אין בנמצא ספרי תרגול מספקים, המציעים לקורא שפע של תרגילים ומאפשרים לו להתמחות במלאכת התכנות. ספר זה נותן מענה למחסור הקיים: מאות תרגילים מגוונים, בנושאי תכנות כלליים ובשפת C בפרט, רוכזו בספר בצירוף פתרונותיהם המלאים. כל העושה את דרכו בתכנות יוכל בעזרת הספר לתרגל את ידיעותיו ולשפר את מיומנות התכנות שלו באופן ניכר.

ייחוד הספר

ספר זה מיוחד הן בשל סוג התרגילים שבו והן בשל מספרם. התרגילים מגוונים ביותר, וכוללים, בין היתר:

- ❖ כתיבת תוכניות מלאות
- ❖ תיקון טעויות במשפטים נתונים
- ❖ מציאת פלט של תוכניות
- ❖ סיווג משפטים נתונים לנכונים / שגויים
- ❖ כתיבת אלגוריתמים ושרטוט תרשימי זרימה

זהו הספר הראשון בעברית המציע שפע כזה של תרגילים. זאת ועוד – כל התרגילים פתורים במלואם, ערוכים על-פי סדר קושי עולה ומחולקים לנושאים ברורים.

בספר כמעט 400 תרגילים, מתוכם 200 עוסקים בכתיבת תוכניות מלאות. כמות עצומה זו של תרגילים מאפשרת למתכנת המתחיל והמתקדם להתמחות בתחומי תכנות רבים.

למי מיועד הספר

הספר מיועד לכל מתכנת מתחיל ומתקדם המעוניין לתרגל את ידיעותיו בתכנות ולשפר אותן. התרגילים ערוכים בסדר עולה ופתורים במלואם (בסוף כל פרק), ולכן ניתן להשתמש בספר הן ללימוד עצמי והן כספר נלווה ללימוד בבית-הספר או בקורס מחשבים אחר.

הספר ערוך במבנהו של ספר הלימוד **שפת C אל השיא** (שיצא בהוצאת הוד-עמי), וכמוהו, הוא מתאים לתלמידים הניגשים לבחינת הברגות במדעי המחשב בשפת C. שני הפרקים הראשונים עוסקים באלגוריתמיקה ובתרשימי זרימה – נושאים חשובים שאינם זוכים לתשומת-לב מספקת בספרי תרגול אחרים.

ספר זה מיועד בעיקר ללומדים שפת C, אך גם המתכנתים בשפות תכנות אחרות יוכלו להיעזר בו. בספר כ-200 תרגילי תכנות מלאים, שאותם ניתן לפתור גם בשפות תכנות אחרות, כדוגמת פסקל, אסמבלי ובייסיק.

ספר הלימוד "שפת C אל השיא"

ספר זה מיועד לתרגול ידיעותיך בתכנות ולא ללימוד החומר. הספר **שפת C אל השיא**, שראה אור בהוצאת הוד-עמי, מכיל את כל החומר התיאורטי הדרוש לפתירת התרגילים שבספר זה. הספר **שפת C אל השיא** ערוך במבנה זהה למבנה של ספר זה, ולכן הוא מתאים במיוחד לחיזוק ידיעותיך בנוגע לתרגילים שבספר.

דרישות

ספר זה הוא כל הדרוש לתרגול החומר ברמה גבוהה ולרכישת מיומנות בתכנות בכלל ובשפת C בפרט. תרגילים רבים בספר דורשים כתיבת תוכניות מלאות בשפת C, ומומלץ להריץ במחשב שברשותך ולהתרשם מפעולתן. מהדר TCLite של חברת Borland מסופק יחד עם הספר, ובאמצעותו תוכל להריץ את כל התוכניות שבספר. אך החשוב מכל הוא כוח הרצון: למידת שפה חדשה, ובוודאי שפת תכנות, אינה משימה קלה; אך אם תלמד בעקביות ותתרגל את החומר הנלמד, תפנים במהרה את חוקי שפת C ותיווכח עד כמה עיסוק זה מהנה ומספק.

שים לב!

אם ברשותך מהדר C, תוכל להשתמש בו במקום במהדר המסופק עם הספר. קיימים מהדרים רבים לשפת C הפועלים תחת סביבות DOS ו-Windows, והתוכניות המופיעות בספר ירוצו בהצלחה תחת כולם.



ארגון הספר

הספר מחולק למספר פרקים, כאשר כל פרק עוסק בנושא אחר. כדי לתכנת ביעילות בשפת C עליך לפתור את רוב התרגילים שבספר – וכל המרבה, הרי זה משובח. התרגילים מגוונים ביותר וערוכים על-פי סדר קושי עולה, וככל שתפתור רבים מהם תעמיק את ידיעותיך בתכנות.

כל התרגילים פתורים במלואם והפתרונות מלווים, במידת הצורך, בהסברים. לנוחיותך, מרוכזים הפתרונות בנפרד בסוף כל פרק, ומומלץ לעיין בהם לאחר פתירת כל תרגיל.

כיצד להשתמש בספר

ניתן לתרגל את הנושאים שבספר בשתי דרכים. אם אינך בקיא בתכנות, מומלץ להתחיל בפרק הראשון ולהתקדם, בקצב המתאים לך, אל הפרקים הבאים. לא מומלץ לדלג על פרקים, משום שרובם מסתמכים על חומר שנלמד בפרקים הקודמים להם. אם אתה כבר מכיר את שפת C, תוכל לתרגל בעזרת הספר נושאים ספציפיים שדורשים חיזוק, תוך שימוש בתוכן העניינים ובאינדקס.

כולי תקווה שספר זה יסייע לך בשיפור ידיעותיך ויכולותיך בתכנות.

~~אשמח לענות על כל שאלה, הערה והארה : zivayalon@email.com~~

זיו איילון,

11/2000

~~התקליטור המצורף~~ ~~והתוכנה TCLite~~

מה בתקליטור

~~התקליטור כולל:~~

- ~~❖ קטלוג HTML - קטלוג ספרי המחשבים האינטראקטיבי של הוצאת הוד-עמי. הקטלוג מאפשר קריאת פרקים לדוגמה, צפייה במידע על ספרי ההוצאה ועוד. לשם קריאת הפרקים לדוגמה יש להתקין את תוכנת Adobe Acrobat Reader אשר נמצאת בתקליטור. הוראות התקנה בתקליטור בקובץ ONCD.~~
- ~~הקטלוג מומלץ לצפייה באמצעות Internet Explorer 5 המצורף בתקליטור. הוראות התקנה בתקליטור בקובץ ONCD.~~
- ~~התקנת שתי התוכנות קלה וניתנת לביצוע באמצעות קישור ישירות מהקטלוג.~~
- ~~❖ מספר תוכנות עזר שימושיות.~~
- ~~❖ תוכנת TCLite וקבצי קוד מקור (קבצי תוכנות) של ספר זה.~~

שים לב!

אם מנהל התקן כונן התקליטורים המותקן הוא 16 סיביות - ייתכן שתראה רק 8 תווים ראשונים של שם הקובץ (במקרה ובמקור הוא ארוך יותר). **הסיבה:** כונני תקליטורים במהירות x4 עובדים עם מנהל התקן שעבד בסביבת DOS ו-Windows 3.11 ויכול לעבוד גם עם Windows 9x, למעט היכולת לזהות קבצים עם שמות ארוכים.

הפתרון: להתקין מנהל התקן 32 סיביות (אם קיים), או לקנות כונן תקליטורים חדש ולוודא שמצורף אליו מנהל התקן 32 סיביות.



~~קרא קובץ ONCD שנמצא בתקליטור לגבי תכולת התקליטור
ולהוראות התקנה של חלק מהתוכנות המצורפות.~~

התיקיות הרלוונטיות לספר

Books\59300 שכוללת את התוכניות, התשובות לשאלות

Software\Tclite שכוללת את המהדר

אם נמצא בידך גם ספר הלימוד "שפת C אל השיא", קבצי התוכניות של ספר זה נמצאים בתיקיה Books\59299.

קוד המקור

בתקליטור נמצאות התשובות שהינן קבצי תוכניות. התוכניות נמצאות בתיקיות לפי מספרי הפרקים ולפי מספרי השאלות בפרק.

העתקת קבצי קוד המקור לדיסק הקשיח

כדי להעתיק את תכולת התיקיה books\59300:

1. לחץ על לחצן התחל, תוכניות, סיוך Windows
 2. לחץ על כונן התקליטורים וסמן את התיקיה 59300 שבתיקיה books.
 3. גרור אותה לכונן C (או אחר) בדיסק הקשיח.
- מכיון שמקור הקבצים הוא התקליטור, הם מסומנים לקריאה בלבד. יש לשנות מאפיין זה כך:

1. בסייר Windows היכנס לתיקיה בדיסק שבה נמצאים הקבצים שהעתקת (c:\59300) או אחרת).
2. סמן את כולם על ידי Ctrl+A.
3. הצב את סמן העכבר מעל האזור המסומן ולחץ לחיצה ימנית בעכבר.
4. מתפריט הקיצור בחר מאפיינים.
5. בטל את הסימון בתיבה קריאה בלבד (דאג שהיא תהיה ריקה).
6. לחץ על החל, לחץ אישור.

בדוק האם השתנה המאפיין על ידי סימון הקובץ: לחיצה ימנית ובחירה במאפיינים בתפריט הקיצור. שים לב שהתיבה של קריאה בלבד תהיה ריקה.

שם לב!

כאשר עובדים במהדר TCLite הכל מתבצע בסביבת המהדר. לא יוצרים קובץ EXE ולא יוצאים ל-DOS כדי להריץ אותו. הספר נכתב לשימוש במהדר רגיל ולכן יש לשים לב לשינויים כאשר עובדים ב-TCLite-ב.



תוכנת TCLite

תוכנת **TCLite** (Turbo C Lite), מהדר לשפת C של חברת Borland International, אשר מיועד למחשבי IBM PC ותואמים, הפועלים תחת מערכת הפעלה MS-DOS. זהו מהדר מהיר מאוד המסוגל, לפי דיווחי החברה, להדר מעל 7,000 שורות בדקה.

התוכנה כוללת:

- ❖ מהדר (compiler).
- ❖ עורך (editor) המשמש לכתיבת ותיקון התוכניות.
- ❖ מערכת תפריטים נוחה, לבחירת האופציות השונות במערכת.
- ❖ מקשר (linker).
- ❖ מערכת איתור שגיאות בהידור.
- ❖ מערכת לבניית תוכניות מקבצי מקור רבים.

אם מותקן כבר מהדר Borland במחשב, אל תתקין את TCLite אלא רק העתק את התוכניות מהתקליטור.

ההתקנה וההפעלה פשוטים. עקוב אחר הסעיפים הבאים.

התקנת התוכנה

הוראות התקנה אלו נמצאות גם בתקליטור בקובץ install.doc (נכתב ב-Word). ההתקנה תבוצע מסביבת DOS ולצורך ההסבר נתייחס לכונן C: (אותן פעולות תעבודנה גם על כל כונן אחר).

<< **אם אתה נמצא ב-Windows - כבה והפעל את המחשב במצב DOS** >>

<< **בחלון כיבוי המחשב סמן את האפשרות**

>> **Restart the computer in MS-DOS mode?**

1. יש לעבור לספריית השורש - על ידי הפקודה: cd \ (קיצור של change directory), ולהגיע למצב זה: _ > C:.

2. במקרה וזה איננו ניסיון התקנה ראשון - יש למחוק את הספרייה TCLITE, בה הותקנה התוכנה. עשה זאת על ידי: deltree tclite, כתוב כך: .C:\>deltree tclite. לבסוף, הקש <Enter>.

אם הספרייה לא קיימת - הסמן יעבור מיידית לשורה חדשה וימתין להוראה הבאה.
אם הספרייה קיימת - תופיע השאלה הבאה:

Delete directory "tclite" and all its subdirectories? [yn] _

כלומר, "האם אתה בטוח שאתה רוצה לבטל ספרייה זו?" הקש על המקש <Y> ואח"כ על <Enter>. המחשב ידווח: Deleting tclite...

3. עתה יהיה עליך לעבור לכונן המתאים של התקליטור (למשל X:). המעבר ל- X: - על ידי הפקודה: **X:** (ואז על המסך יהיה כתוב: C:\>X:). לאחר הקשת <Enter> הסמן המהבהב יופיע אחרי X:\>_.

4. עבור לספרייה software\tclite על ידי הקלדת הפקודה:

X:\>cd software\tclite

5. עכשיו הכל מוכן לביצוע ההתקנה. כתוב את הפקודה:

X:\software\tclite>install C:

(שפירושה, התקנה לתוך כונן C:). שים לב: אם תשכח לכתוב C:, כלומר לאן יש לבצע את ההתקנה - פשוט חזור שוב על הפקודה עם 2 חלקיה). המחשב ידווח, הן על התקנת התוכנה TCLITE - גרסה של שפת Turbo C++ . משך זמן ההתקנה תלוי במהירות המחשב שלך, ויכול להימשך אפילו מעל 2 דקות.

6. במקרה של תקלה (תקיעת המערכת) יש לבטל את ההתקנה ע"י <Ctrl>+<C>. במקרה זה המחשב ישאל: Terminate batch job (Y/N)?, כלומר, האם לסיים את העבודה. יהיה עליך להקיש <Y> ו- <Enter>, ואז, בהנחה שיש די זיכרון בדיסק הקשיח, לחזור על שלבים 2-5.

7. לאחר סיום ההתקנה (והעתקת קבצי קוד המקור) שים את התקליטור בצד ושמור אותו כגיבוי. הפעלת התוכנית והרצת התוכניות נעשית מתוך הדיסק הקשיח.

הפעלת התוכנה

1. יש לחזור מכונן X: לכונן C:, ולהימצא בספריית השורש, על ידי הפקודה C: והקשת Enter. אם אינך נמצא ב- C:\> הקש cd\ ו-Enter.

2. הכניסה למהדר על ידי הפקודה C:\>c המפעילה את קובץ ההפעלה C.BAT. כתוצאה, מקבלים את סביבת העבודה של Turbo C/C++, ובתוכה חלון פתוח בצבע כחול, ששמו: **NONAME00.CPP** (רק בפעם הראשונה שנכנסים לתוכנה).

3. כדאי לסגור את החלון **NONAME00.CPP** ע"י צמד המקשים <Alt>+<F3>, ואז מקבלים חלון אפור ריק, שבראשו רשימת תפריטים על רקע לבן. כרגע אתה מוכן להתחיל לעבוד בעזרת תוכניות הספר.

שים לב!

ברירת המחדל של ה- Compiler היא לקרוא ולשמור קבצי C/CPP. אתם כמובן קוראים קבצי C וכותבים קבצי C.



הרצת תוכניות

1. כדי לקבל תוכנית - יש להקיש <F3>. הקשה זו פותחת "חלון שיחה", שכותרתו הראשית Load a File ("טען קובץ") וכותרת משנה ראשונה הינה: Name ואחריה, בתוך שורת כחולה עם הדגשה ירוקה כתוב: *.CPP. אם 5 תווים אלה (*.cpp) אכן מופיעים - הקש Backspace פעמיים כדי שיישאר רק *.c. אם אינך רואה דבר הקלד *.c, ולאחר מכן הקש <Enter>.
2. תעבור לחלון פנימי שכותרתו Files ("קבצים"), ועל רקע בצבע תכלת מצויים 2 טורים של שמות קבצים. באופן כללי, מקשי החיצים מאפשרים לטייל עם ה"סמן", שצבעו ירוק בהיר, לכל הקבצים.
3. למעשה, מה שמוצג כרגע על המסך הוא שמות הקבצים/תוכניות שבספרייה programs. הקשה על <Enter> תכניס אותך לתוכנית. כדי לבחור ספרייה אחרת (mesimot או targilim) הקש על \. שמופיע בסוף הרשימה של התוכניות.
4. הקשת <Enter> על שם קובץ תפתח חלון כחול, שכותרתו במרכז הינה: שם הקובץ, ובתוך החלון יופיע הטקסט של תוכנית זו (או של קובץ זה).
5. כדי להריץ את התוכנית עליך להקיש על צמד המקשים: <Ctrl>+<F9>. המחשב יתרגם את התוכנית לשפת מכונה ויבצע אותה. כדי לראות את פלט התוכנית הקש <Alt>+<F5>. יש להקיש על מקש כלשהו כדי לחזור לחלון התוכנית.
6. המשך ההרצה של תוכניות נוספות מומלץ באופן הבא: תחילה לסגור את חלון התוכנית הנוכחית ע"י צמד המקשים <Alt>+<F3>. אח"כ ניתן לחזור על השלבים 1 עד 5, כאשר בשלב 3 אפשר לבחור תוכנית אחרת.

שים לב!

בכל שלב בסעיפי הרצת התוכניות אפשר לבטל שלב כלשהו ולחזור לשלב הקודם ע"י המקש <Esc>.



זהירות!

אם המחשב שלכם מחובר לרשת, המלצתנו - בזמן אתחול המערכת להקיש F5 כדי לדלג על ביצוע הקבצים autoexec.bat ו-config.sys אם לא תעשו כן, ייתכן שבתוכניות תראו תווים לא מובנים במקום עברית.



עקרונות העבודה


שיטת העבודה במערכת הפיתוח מבוססת על **מערכת תפריטים** (Menus) שמתוכם בוחרים באפשרויות השונות. בהתאם לבחירה, מופיעים זה על גבי זה, חלונות המשמשים לקליטת נתונים מהמשתמש ולהצגת אפשרויות נוספות.

קבלת עזרה (HELP), מקש F1

בכל עת ניתן לקבל מהמערכת דברי הסבר (באנגלית) המתאימים לשלב העבודה שבו נמצאים. **בהקשת F1**, המערכת תגיב בהקרנת חלון המכיל דברי הסבר על מקום הימצאך והאפשרויות העומדות לרשותך באותו שלב עבודה. ייתכן ותופענה מילות מפתח אשר תאפשרנה לקבל הסברים נוספים הקשורים לאותו נושא. כאשר תציב את הסמן על אחת מהמילים האלו ותקיש **Enter**, יוצג ההסבר המתאים.

כל הקשה על **Alt+F1** תחזיר אותך לחלון העזרה הקודם. הקשה על **F1** בזמן שמוקרן חלון עזרה כלשהו, תגרום למערכת להקרין חלון עזרה ראשי, שממנו תוכל לבחור נושא כלשהו ולקבל עליו חלון הסבר.

שים לב!
כאשר ברצוננו לקבל עזרה אודות פונקציה מסוימת, יש להקליד את שם הפונקציה, לדאוג שהסמן יעמוד על הפונקציה, ולהקיש Ctrl+F1 . כך נקבל עזרה מותאמת לפונקציה ובדרך כלל תוכנית דוגמה, המסבירה את השימוש בפונקציה.



מקש Esc משמש ליציאה מחלונות העזרה בחזרה לנקודה ממנה נכנסת אליהם.

התפריט הראשי - Main Menu

לתפריט הראשי ניתן להגיע מייד לאחר הרצת טורבו C, או על ידי הקשה על **F10** מנקודה כלשהי בזמן העבודה במערכת. עובדה זו (F10-Main Menu) מצוינת בשורה התחתונה בכל עת.

בזמן היותך בתפריט הראשי, אחד הפריטים (בשורה עליונה) יהיה מודגש (מואר). ביכולתך להציב את הסמן על פריט אחר באמצעות מקשי החיצים ולהקיש **Enter**, או שתוכל להקיש את האות הראשונה של כל פריט (לדוגמה F - עבור File).

אם אינך נמצא בתפריט הראשי תוכל לבחור פריט מסוים על ידי הקשת **Alt** והאות הראשונה של הפריט (לדוגמה **Alt+F** - עבור File).

לפניך תיאור קצר של הפריטים בתפריט הראשי:


File טיפול בקבצים, טעינה, שמירה, החלפת כונן ו/או מדרוך ברמת DOS, יציאה זמנית ל-DOS וסיום העבודה עם TCLite.

Edit מצב עריכה של התוכנית הנמצאת בחלון העריכה (העליון).

חיפוש והחלפת תווי טקסט בקובץ התוכנית.	Search
הרצת תוכנית, לאחר ביצוע הידור וקישור.	Run
הפעלת שלבי ההידור והקישור.	Compile
הרצת התוכנית צעד אחר צעד, קביעת נקודות עצירה, והצצה בערכי משתנים בזמן ריצה.	Debug
הגדרת הידור עבור תוכנית שקטעים שלה נמצאים במספר קבצי מקור.	Project
הצגת האפשרויות השונות הקיימות במערכת הפיתוח המוכללת. טיפול במיקום קבצי הספרייה וקבצי ההגדרות.	Options
טיפול בחלונות המסך השונים.	Window
מערכת עזרה לסביבת הפיתוח.	Help

שים לב!

מומלץ ללמוד את הקיצורים השונים המצוינים בתפריטים כדי לחסוך את זמן הגישה לתפריטים.



ענה נבצע סקירה כללית של מירב האפשרויות במסגרת התפריטים אשר תוארו לעיל. הסברים נוספים ניתן למצוא במערכת העזרה. צירופי המקשים אשר יופיעו בסוגריים יבצעו ישירות את פעולת התפריט.

תפריט File

לאחר הבחירה תפתח תיבת דו-שיח אשר לתוכה ניתן להקיש שם תוכנית כולל המסלול המלא שלה, אם צריך. על ידי הקשת Tab ניתן לעבור לחלקים השונים של תיבת הדו-שיח.	Open (F3)
פתיחת חלון לתוכנית חדשה אשר רק על ידי שמירה (Save) ומתן שם לתוכנית תיבצר קובץ חדש.	New
אפשרות זו מיועדת לשמירת הקובץ המצוי בחלון העריכה. אם עדיין לא ניתן שם לקובץ, תיפתח תיבת דו-שיח שבה ניתן להקיש את שם הקובץ הרצוי.	Save (F2)
שמירת קובץ קיים בשם אחר.	Save as
הקובץ הקודם לא יימחק. על הדיסק נמצאים עכשיו שני קבצים זהים לחלוטין, אך בעלי שם שונה.	

שמירת כל הקבצים הפתוחים.	Save all
שינוי כוונן/ספריית העבודה. יש לבחור ב-"OK" בתיבת הדו-שיח כדי שהשינוי יתבצע.	Change dir
הזפסת התוכנית למדפסת.	Print
הצגת חלון מידע אשר מכיל פרטים אודות התוכנית והזיכרון שבשימוש.	Get info
יציאה זמנית לסביבת DOS. על ידי הקלדת Exit ניתן לחזור לסביבת הפיתוח.	DOS shell
סיום הפעילות ב-TCLite והחזרת הפיקוח למערכת ההפעלה.	Quit (Alt+x)
תפריט Edit	
שיחזור שורת קוד לאחר שינוי או מחיקה.	Restore line
גזירת קטע קוד מסומן (בלוק) אל תוך ה-clipboard.	Cut (Shift+Del)
העתקת קטע קוד מסומן (בלוק) אל תוך ה-clipboard.	Copy (Ctrl+Ins)
הדבקת קטע קוד אשר הועתק או נגזר אל תוכנית במקום שעומד הסמן.	Paste (Shift+Ins)
העתקת תוכנית דוגמה אשר מופיעה ב-Help אל תוך התוכנית.	Copy example
מראה מה שיש בלוח.	Show clipboard
מחיקת קטע קוד מסומן.	Clear (Ctrl+Del)
פעולות עריכה בחלון הפיתוח	
תנועת הסמן לכל הכיוונים.	חיצים
קפיצת דף מסך מטה/מעלה.	Page Up/Page Down
עבור לתחילת התוכנית.	Ctrl+Page Up
עבור לסוף התוכנית.	Ctrl+Page Down
מחיקת שורה.	Ctrl+y
מחיקת תו קודם.	backspace (←)
מחיקת תו נוכחי (התו שהסמן נמצא מעליו).	Delete
סימון קטע קוד (הקטע הרצוי יואר בצבע אפור בהיר).	חיצים+Shift

תפריט Search

תיבת דו-שיח לחיפוש, בה יש להקליד מחרוזת לחיפוש. בחירת "OK" תאשר את ביצוע החיפוש.	Find
בחירת מחרוזות לחיפוש ומחרוזות להחלפה בתיבת הדו-שיח. חזור על פעולת החיפוש/החלפה האחרונה.	Replace
	Search again (Ctrl+I)
עבור לשורה על ידי הקלדת השורה הנבחרת בתיבת הדו-שיח. ועוד...	Go to line number

תפריט Run

ביצוע הידור לתוכנית, והרצה מתוך סביבת הפיתוח (ניתן לראות את פלט התוכנית על ידי הקשת Alt+F5).	Run (Ctrl+F9)
הפסקת ריצת התוכנית כאשר נמצאים בריצת צעד-צעד (שורה אחר שורה). אפשר גם להקיש F7.	Program reset (Ctrl+F2)
הרצת חלק מהתוכנית, עד למקום הסמן.	Go to cursor (F4)
הרצת התוכנית שורה אחר שורה, כולל כניסה לפונקציות.	Trace into (F7)
הרצת התוכנית שורה אחר שורה, לא כולל כניסה לפונקציות.	Step over (F8)
העברת ארגומנטים לתוכנית כאשר עובדים עם שורת הפקודה.	Arguments

תפריט Compile

בצע הידור לתוכנית.	Compile to OBJ (Alt+F9)
צור קובץ ריצה .exe.	Make EXE file (F9)
בצע הידור והרץ את התוכנית.	Link EXE file (Ctrl+F9)
בצע הידור לכל הקבצים המשתתפים בפרויקט.	Build all

תפריט Debug

פתיחת חלון בו ניתן לראות את ערך המשתנה בתהליך ריצה, שלב אחר שלב.	Inspect (Alt+F4)
פתיחת חלון בו ניתן לראות ולשנות ערך של משתנה בתהליך ריצה, שלב אחר שלב.	Evaluate/modify (Ctrl+F4)
הצצה אל תוכן המחסנית.	Call stack (Ctrl+F3)

חלון בו ניתן לצפות בערך המשתנים בזמן ריצה (שלב אחר שלב).	watches \ Add watch (Ctrl+F7)
קביעת נקודות עצירה בתוכנית (שורה מוארת בצבע אדום). התוכנית תרוץ עד לנקודת העצירה הראשונה במשך ריצת התוכנית. הקשה על Ctrl+F9 תביא אותנו לנקודת העצירה השנייה, וכך הלאה עד לסוף התוכנית. קביעת נקודת עצירה על שורה שכבר קבענו לה נקודת עצירה, תבטל את נקודת העצירה.	Toggle breakpoint (Ctrl+F8)

תפריט Project

פתיחת פרויקט חדש.	Open project
סגירת פרויקט.	Close project
הוספת קובץ מקור לפרויקט.	Add item
מחיקת קובץ מקור מהפרויקט.	Delete item

תפריט Options

קביעת מסלולים לספריית המערכת ולספריית העבודה שניצור תחת TCLite.	Directories
קביעת תכונות סביבת העבודה: העדפות, סביבת העריכה ומצב העכבר.	Environment

תפריט Window

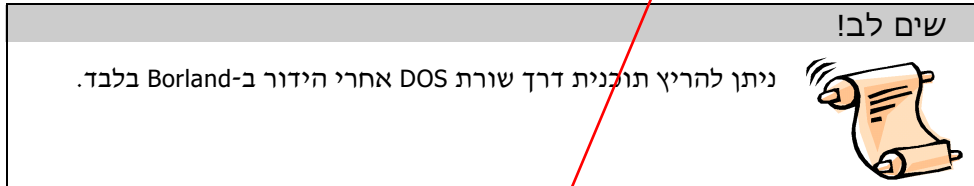
מסגרת החלון מוצגת בצבע ירוק ובעזרת חיצים + Shift ניתן לשנות את גודל חלון העבודה. בעזרת החיצים בלבד ניתן גם להזיז את חלון העבודה על פני המסך.	Size/Move (Ctrl+F5)
הגדלת חלון העבודה למסך מלא. בהקשה נוספת יחזור החלון לגודלו הקודם.	Zoom (F5)
דילוג בין החלונות הפתוחים.	Next (F6)
סגירת חלון.	Close (Alt+F3)
הצגת חלון ריצת התוכנית.	Output
מעבר אל חלון ריצת התוכנית.	User screen (Alt+F5)
הצגת רשימת החלונות הפתוחים בתיבת דו-שיח.	List (Alt+0)

חלון העריכה

בכל פעם שיוצרים קובץ חדש, או כשפותחים קובץ קיים נפתח חלון העריכה.

הרצת תוכנית דרך שורת DOS

בכל פעם ששלב ההידור-קישור עובר ללא כישלון, נוצר קובץ הנושא את שם קובץ התוכנית עם סיומת EXE. זהו קובץ הריצה. כדי לצאת ל-DOS הקש Alt+X, או בחר בפריט **Quit** שבתפריט **File**.



בשורת DOS הקש את הפקודה: * שם קובץ dir ותוכל לראות שקיימים 4 קבצים בעלי שם כזה, הנושאים סיומות שונות:

.BAK מצב הקובץ לפני עדכונים בחלון העריכה (בתנאי שהמתג Backup Source Files מופעל).

.C קובץ התוכנית.

.EXE קובץ הריצה.

.OBJ קובץ זמני של המהדר (שלב ביניים לפני הקישור לספריות המערכת).

נחזור ונזכיר, שפרק זה מהווה מבוא תמציתי למהדר TCLite ונועד לשרת את התרגול בשפת C בעת הלימוד בספר בלבד.

כתיבת עברית בתוכנית

כדי לכתוב עברית בתוכנית ב-TCLite:

1. העתק את הקובץ `heb.com` מהספרייה בתקליטור `software\tclite` לספרייה הראשית בדיסק הקשיח (בדרך כלל C).
2. הקש במצב DOS - `heb` >C:\ ו-Enter.
3. היכנס ל-tclite וכאשר תרצה להקליד עברית, הקש — Ctrl+ (החזק את Ctrl ולחץ על מקש מינוס). נפתח חלון בו תוכל לכתוב. הקש Enter כדי להכניס את מה שהקלדת לתוכנית.

קרא קובץ ONCD שנמצא בתקליטור לגבי תכולת התקליטור ולהוראות התקנה של חלק מהתוכנות המצורפות.

מה עוד בתקליטור?

הוצאת הוד-עמי מפיצה תוכנות אלו כבנוס ללקוחות ההוצאה, ואינה מתיימרת לגבות תשלום עבור התוכניות המצורפות ו/או לתמוך בהם.

שים לב!

השימוש בתקליטור זה הוא על אחריותו הבלעדית של המשתמש. המוצרים המותקנים בתקליטור זה מסופקים באחריות החברות המייצרות אותם. הוצאת הוד-עמי אינה אחראית, בכל צורה שהיא, לאופן ולטיב התוכנות המותקנות.



בכל שאלה לגבי תוכנה הנמצאת בתקליטור, יש לפנות למפתחי התוכנה (כל תוכנה בנפרד) כפי שמצוין בקבצי העזרה של התוכנה המדוברת.

הקבצים הם גרסאות ShareWare ו-FreeWare.

גרסת ShareWare מאפשרת לך, המשתמש, לבדוק את יעילות התוכנה ואת תאימותה לעבודה אותה מביצע. אם נמצאה התוכנה מתאימה לצרכיך, עליך לשלם למפתחיה תשלום סמלי (לפי הרשום בקבצי העזרה של כל תוכנה ותוכנה בנפרד) כדי לקבל רישיון מלא לשימוש בה. קבלת רישיון לשימוש בתוכנה יפתח בפניך מיגוון אפשרויות שלא עמדו לרשותך בהפעלת גרסת ה-ShareWare.

פרק 1

אלגוריתמיקה

שאלות

מושגי תכנות בסיסיים

1. הקף בעיגול את המשפטים הנכונים:
 1. כל בעיה ניתנת לפתירה על ידי מחשב משוכלל וחדיש.
 2. בעיה שניתן לפותרה בשלבים ברורים וחד-משמעיים מכונה **בעיה לוגית**.
 3. מתכנת נוהג להגדיר למחשב רצף הוראות הפותר בעיה לוגית נתונה.
 4. המתכנת מתקשר עם המחשב באמצעות שפת תכנות.
 5. אין צורך למלא אחר כל חוקי שפת התכנות כדי שהמחשב יבצע את הוראותיה.
 6. המחשב נוהג לקרוא את תוכנית המחשב ולבצע את הוראותיה.

האלגוריתם

כללים בתכנון האלגוריתם

2. הקף בעיגול את המשפטים הנכונים:
 1. המחשב יתקן בעצמו הוראה שגויה הניתנת לו אם הטעות הינה מזערית.
 2. גם אם ההוראות הניתנות למחשב נכונות מבחינה תחבירית, תיתכן תקלה בעת ריצת תוכנית.
 3. האדם או המכונה אליהם מיועדות ההוראות חייבים להבין כיצד לבצען.

4. רצף הוראות נקי מטעויות תחביריות הנשלח למחשב יבצע תמיד את אשר ציפינו ממנו.
5. המחשב ידחה כל הוראה שגויה מבחינה תחבירית הניתנת לו.
6. המחשב ידחה כל הוראה שגויה מבחינה לוגית הניתנת לו.

אלגוריתמים המיועדים לאדם

3. לפניך רצפי הוראות המנחים אישים שונים כיצד לבצע מטלות שונות. הקף את הרצפים השגויים וציין את השגיאה בהם.

1. רצף הוראות המורה לאדם כיצד להתרחץ :

(1) היכנס לאמבטיה

(2) פתח את ברז המים

(3) הורד את בגדיך

2. רצף הוראות המורה לאדם כיצד לבלוע קפסולת אקמול :

(1) מלא כוס במים

(2) הנח את הקפסולה על לשונך

(3) לגום מהמים ובלע את הקפסולה

3. רצף הוראות המורה לאדם כיצד לחפוף את ראשו :

(1) לחץ על בקבוק השמפו

(2) מרח את הנוזל על גופך

(3) רחץ את גופך

4. רצף הוראות המורה לאדם כיצד לצלם את חברו :

(1) הדלק את המצלמה

(2) לחץ על כפתור הצילום

(3) כבה את המצלמה

5. רצף הוראות המורה למזכירה כיצד להדפיס מכתב במדפסת שברשותה :

(1) הקלידי את המכתב במעבד תמלילים

(2) הדליקי את המחשב

(3) הדליקי את המדפסת

(4) בצעי פעולת הדפסה מהמחשב

6. רצף הוראות המורה לאדם כיצד לנעול את נעלו :

(1) הכנס את כף רגלך אל הנעל

(2) פרום מעט את שרוכי הנעל

(3) קשור את שרוכי הנעל

7. רצף הוראות המורה לנהג כיצד לצאת מחנייה :

(1) הדלק את רכבך

(2) אותת כנדרש

(3) לחץ על דוושת הגז בהדרגה

8. רצף הוראות המורה לצרכן כיצד להתקין את המשחק החדש שקנה :

(1) פתח את אריזת המשחק

(2) הפעל את קובץ ההתקנה מהתקליטון

(3) הכנס את התקליטון לכונן התקליטונים

4. כתוב אלגוריתמים בני 3-4 שלבים המכילים הוראות לאישים הבאים :

1. אדם המחפש עבודה בעיתון "דרושים".

2. אדם הקונה פחית שתייה ממכונת מזון אוטומטית.

3. אדם המשכיר סרט בוידאומט.

4. עובר אורח המעוניין להגיע לתחנת האוטובוסים הקרובה.

5. תלמיד העומד להיבחן בבחינה בכתב.

6. זאטוט העומד לחצות את הכביש.

שים לב!

את האלגוריתמים המוצגים למעלה ניתן לכתוב בכמה דרכים.
אפשרות פתרון אחת מוצגת בתשובות.



אלגוריתמים פשוטים

הוראת הקלט

חוקים במתן שמות למשתנים

5. אחדים משמות המשתנים שלפניך שגויים. סמן אותם והסבר את הטעויות.

- | | | | |
|----|-------------|-----|---------------|
| 1. | population | 6. | benandjerrys |
| 2. | 98salary | 7. | [brackets] |
| 3. | salary98 | 8. | מחוצע |
| 4. | c_plus_plus | 9. | _acceleration |
| 5. | c++ | 10. | ben&jerry's |

6. ציין באילו משמות המשתנים הבאים כדאי להשתמש בתוכניות מורכבות:

- | | | | |
|----|---------|-----|-------------|
| 1. | average | 6. | biggest_num |
| 2. | a | 7. | something |
| 3. | x | 8. | address |
| 4. | length | 9. | number |
| 5. | xyz | 10. | bluesky |

הוראת הפלט / הוראת ההשמה

7. ציין את השגיאה בכל אחת מהוראות ההשמה שלפניך:

1. $x \leftarrow 2, y \leftarrow 3$
2. $weight \rightarrow 59$
3. $x + y \leftarrow 7$
4. $a \leftarrow b \leftarrow c$
5. $sum = 5$

8. תחום כל אחד מהביטויים הבאים בסוגריים מתאימים (על-פי סדר קדימויות האופרטורים) ומצא את ערכו :

- | | |
|--|--|
| <p>1. $1 + 2 * 3$</p> <p>2. $2 * 4 \% 5$</p> <p>3. $2 ^ 3 + 5$</p> <p>4. $5 \% 3 * 4 \% 3$</p> <p>5. $1 + 3 \% 4 ^ 2$</p> | <p>6. $4 * 2 ^ 2 ^ 3$</p> <p>7. $7 \% 9 ^ 3 * 3$</p> <p>8. $3 * 4 / 3 * 6 \% 5$</p> <p>9. $3 ^ {(\sqrt{15\%10 * 5} - 3)}$</p> <p>10. $\sqrt{4^3 - 2 * 5 * 4 + 1}$</p> |
|--|--|

9. כתוב את האלגוריתמים הבאים :

- אלגוריתם הפולט את השטח של מעגל, שאת רדיוסו מזין המשתמש.
- אלגוריתם הפולט את השטח של מלבן, שאת אורכי צלעותיו מזין המשתמש.
- אלגוריתם הפולט את אורך היתר במשולש ישר-זווית, שאת אורכי ניצביו מזין המשתמש.

10. כתוב את האלגוריתמים הבאים :

- אלגוריתם הפולט את הממוצע של שלושה ציונים שמזין המשתמש.
- אלגוריתם הפולט את הממוצע הגיאומטרי של שלושה מספרים שמזין המשתמש (הממוצע הגיאומטרי של המספרים a , b ו- c הוא $\sqrt[3]{a * b * c}$).
- אלגוריתם הפולט את שווי המס שיש לשלם על משכורת נתונה. את המשכורת ואת שער המס (באחוזים) מזין המשתמש.

תנאים

ביצוע מותנה באלגוריתמים המיועדים למחשב

11. מצא אילו מהביטויים הבאים מתקיימים ואילו לא. התייחס לערך המשתנה a כאל 8 ולערך המשתנה b כאל 5.

- | | |
|--|--|
| <p>1. $3 \leq 2$</p> <p>2. $5 * 3 \geq 3 * 5$</p> <p>3. $4 * 4 <> 4 + 4$</p> <p>4. $4 - 7 + 1 > 4 * (7 + 1)$</p> <p>5. $5 * 2 = 2 + 1 * 8$</p> | <p>6. $2 * 2 + 2 = 8$</p> <p>7. $a * a \leq b * b$</p> <p>8. $b \% a <> 3$</p> <p>9. $b + a * 3 \leq (b + 4) / a$</p> <p>10. $b > (a ^ (a - a))$</p> |
|--|--|

12. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הקולט מספר מהמשתמש ופולט את המלל הבא : **זוגי** אם המספר זוגי ; **חיובי** אם המספר חיובי ; **גדול** אם המספר גדול מ-100.
2. אלגוריתם הקולט שלושה מספרים מהמשתמש, כאשר המספר השלישי הוא המספר שהמשתמש משער שהוא סכום שני המספרים הראשונים. תיפלט הודעה האם המספר השלישי שהוזן הוא אכן סכום שני המספרים הראשונים.
3. אלגוריתם הקולט שני מספרים שונים מהמשתמש ופולט אותם על המסך על-פי גודלם (המספר הגדול ייפלט ראשון, והקטן – שני).

התנאי המורכב וגם / התנאי המורכב או

13. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הקולט מהמשתמש מספר שלם ופולט **מתחלק** אם הוא מתחלק ב-30 ללא שארית, על ידי שימוש במספרים 2, 3 ו-5 בלבד.
2. אלגוריתם הקולט מהמשתמש שלושה מספרים ופולט **שונים** אם הם אינם זהים.
3. אלגוריתם הקולט מהמשתמש שלושה מספרים ופולט **מתאים** אם הם מהווים שלשה פיתגוראית (כלומר, שריבוע אחד המספרים שווה לסכום ריבועי המספרים האחרים).

הוראות תנאי מקוננות

14. הפוך את האלגוריתם הראשון שכתבת בתרגיל 13 לאלגוריתם המכיל הוראות תנאי מקוננות.

לולאות

בצע ... פעמים

15. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הקולט מהמשתמש 5 מספרים ופולט את מכפלתם זה בזה.
2. אלגוריתם הקולט מהמשתמש 20 מספרים ופולט כמה מהם שליליים.
3. אלגוריתם הקולט מהמשתמש 10 מספרים חיוביים ופולט את המספר הגדול ביותר שהוזן.

16. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הפולט את n המספרים הטבעיים הראשונים המתחלקים ב-3 ללא שארית, כאשר n הוא מספר טבעי שמזין המשתמש.
2. אלגוריתם הפולט את סכום המספרים השלמים מ-1 ל- n (כולל), כאשר n הוא מספר טבעי שמזין המשתמש.
3. אלגוריתם הפולט את מכפלת n המספרים הזוגיים החיוביים הראשונים (כלומר, $2 * 4 * 6 * 8 * \dots$), כאשר n הוא מספר טבעי שמזין המשתמש.

עבור ... מ-... עד ... בצע

17. כתוב את האלגוריתמים בתרגיל 16 על ידי לולאות מסוג "עבור ... מ-... עד ... בצע".

18. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הפולט את $n!$ (העצרת של n , כלומר הביטוי $n * (n-1) * (n-2) * \dots * 1$), כאשר n הוא מספר טבעי שמזין המשתמש.
2. אלגוריתם הפולט את סכום הסדרה הבאה (n הוא מספר טבעי שמזין המשתמש):

$$n^1 + n^2 + n^3 + \dots + n^n$$

3. אלגוריתם הפולט את n האיברים הראשונים (n הוא מספר טבעי שמזין המשתמש) של הסדרה הבאה :

$$\frac{1}{n}, \frac{1}{n+1}, \frac{1}{n+2}, \dots, \frac{1}{n+n}$$

כל עוד ... בצע

19. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הקולט מהמשתמש שני מספרים טבעיים, $from$ ו- to , כאשר $from < to$. האלגוריתם יפלוט את כל המספרים השלמים הנמצאים בין $from$ ו- to (כולל אותם) מהסוף להתחלה.
2. אלגוריתם הקולט מהמשתמש שני מספרים טבעיים, a ו- b , ופולט את החזקה שלהם (a^b) ללא שימוש באופרטור החזקה ($^$).
3. אלגוריתם הקולט מהמשתמש מספר שלם גדול מ-1 ופולט 1 אם הוא מספר ראשוני ו-0 אם לאו. מספר ראשוני הוא מספר המתחלק ללא שארית רק ב-1 ובעצמו.

בצע ... עד ש ...

20. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הקולט מהמשתמש שני ציונים, שוב ושוב, עד שהם נמצאים בטווח הנכון; כלומר, בין 0 ל-100 (כולל). לבסוף יפלוט האלגוריתם את ממוצע הציונים.
2. אלגוריתם הקולט מהמשתמש שני מספרים טבעיים, ומבקש מהמשתמש למצוא את סכום השורשים של שני המספרים. המשתמש יזין את תשובתו שוב ושוב, עד שיצדק.
3. אלגוריתם הקולט מהמשתמש מספר טבעי ופולט את סכום ספרותיו.

אלגוריתמים ומערכים

21. כתוב את האלגוריתמים הבאים :

1. אלגוריתם הקולט מספר מהמשתמש ומציב אותו בתא הראשון של מערך בן 20 תאים. החל מהתא השני, יוצב בכל תא במערך האינדקס שלו, ויתווסף אליו המספר שבתא הקודם לו.
2. אלגוריתם המציב בשני התאים הראשונים של מערך בן 15 תאים את הערכים 0 ו-1, בהתאמה, והחל מהתא השלישי, מציב בכל תא את סכום המספרים שבשני התאים הקודמים לו.
3. אלגוריתם הקולט מהמשתמש 10 מספרים, מאחסן אותם במערך מתאים, וכופל כל מספר באינדקס של התא בו הוא נמצא. לבסוף, יפלוט האלגוריתם את המספרים שבמערך, מהסוף להתחלה.

תשובות

.1

.4 ✓	.1 x
.5 x	.2 ✓
.6 ✓	.3 ✓

.2

.4 x	.1 x
.5 ✓	.2 ✓
.6 x	.3 ✓

.3

.5 x על שלב (2) להופיע לפני שלב (1)	.1 x על שלב (3) להופיע לפני שלב (1)
.6 x על שלב (2) להופיע לפני שלב (1)	.2 ✓
.7 ✓	.3 x שלבים (2)-(3) אינם מתאימים לחפיפת ראש, בה מוקצף השמפו על השיער
.8 x על שלב (3) להופיע לפני שלב (2)	.4 ✓

<p>4. (1) פנה שמאלה בפנייה הבאה (2) המשך לצעוד עד לגן של בית מספר 8 (3) פנה ימינה וצעד כ-200 מטרים (4) חצה את הכביש</p>	<p>1. (1) פתח את עיתון ה"דרושים" (2) עבור לקטגוריית העבודות המעניינת אותך (3) בחר מודעה מתאימה (4) צור קשר עם מפרסם המודעה</p>
<p>5. (1) קרא את השאלון רק לאחר קבלת אישור מהמשיח (2) כתוב בעט ובכתב ברור ונקי (3) עם תום הבחינה החזר את השאלון למשיח</p>	<p>2. (1) בחר את המשקה המבוקש (2) שלשל מטבעות כנדרש לחריץ המתאים (3) לחץ על הכפתור המתאים למשקה המבוקש (4) קח את פחית השתייה</p>
<p>6. (1) עמוד על שפת המדרכה (2) הסתכל היטב לשני הצדדים (3) כאשר הכביש פנוי, חצה אותו בזהירות</p>	<p>3. (1) העבר את כרטיס האשראי בחריץ המתאים (2) בחר את הסרט המבוקש (3) לחץ על הכפתור המתאים לסרט המבוקש (4) קח את קלטת הווידאו</p>

.6 ✓	1. ✓
7. ✗ שם משתנה אינו יכול להכיל את התווים [ו-]	2. ✗ אין להתחיל שם משתנה בספרה
8. ✗ שם משתנה אינו יכול להכיל אותיות עבריות	3. ✓
9. ✓	4. ✓
10. ✗ שם משתנה אינו יכול להכיל את התווים & ו- '	5. ✗ שם משתנה אינו יכול להכיל את התו +

.6

.6 ✓	.1 ✓
.7 ✗	.2 ✗
.8 ✓	.3 ✗
.9 ✓	.4 ✓
.10 ✗	.5 ✗

.7

1. אין להציב ערכים ביותר ממשתנה אחד באותה הוראת השמה
2. על החץ בהוראת ההשמה לפנות שמאלה
3. משמאל לחץ חייב להופיע שם משתנה אחד, ולא ביטוי מתמטי
4. בהוראת ההשמה חייבים להופיע שני שדות בלבד, המופרדים על ידי חץ בודד
5. יש להשתמש בחץ לביצוע פעולת ההשמה

.8

1.	$1 + (2 * 3) = 7$	6.	$4 * ((2 ^ 2) ^ 3) = 256$
2.	$(2 * 4) \% 5 = 3$	7.	$(7 \% (9 ^ 3)) * 3 = 21$
3.	$(2 ^ 3) + 5 = 13$	8.	$(((3 * 4) / 3) * 6) \% 5 = 4$
4.	$((5 \% 3) * 4) \% 3 = 2$	9.	$3^{\sqrt{(159\%10)*5}} - 3 = 9$
5.	$1 + (3 \% (4 ^ 2)) = 4$	10.	$\sqrt{((4^3) - ((2*5)*4)) + 1} = 5$

.9

.1

(1) קלוט מספר למשתנה r

$$\text{area} \leftarrow \pi * r * r (2)$$

(3) הדפס את הערך של area

.2

(1) קלוט מספר למשתנה length

(2) קלוט מספר למשתנה width

$$\text{area} \leftarrow \text{length} * \text{width} (3)$$

(4) הדפס את הערך של area

.3

(1) קלוט מספר למשתנה a

(2) קלוט מספר למשתנה b

(3) $c \leftarrow \sqrt{a^2 + b^2}$

(4) הדפס את הערך של c

.10

.1

(1) קלוט מספר למשתנה grade1

(2) קלוט מספר למשתנה grade2

(3) קלוט מספר למשתנה grade3

$average \leftarrow (grade1 + grade2 + grade3) / 3$

(5) הדפס את הערך של average

.2

(1) קלוט מספר למשתנה a

(2) קלוט מספר למשתנה b

(3) קלוט מספר למשתנה c

(4) $average \leftarrow \sqrt[3]{a * b * c}$

(5) הדפס את הערך של average

.3

(1) קלוט מספר למשתנה salary

(2) קלוט מספר למשתנה tax_rate

(3) $tax \leftarrow salary * tax_rate / 100$

(4) הדפס את הערך של tax

.11

.6 x	.1 x
.7 x	.2 ✓
.8 ✓	.3 ✓
.9 x	.4 x
.10 ✓	.5 ✓

.12

.1

(1) קלוט מספר למשתנה num

(2) אם $num \% 2 = 0$ אזי

(2.1) הדפס את ההודעה "זוגי"

(3) אם $num > 0$ אזי

(3.1) הדפס את ההודעה "חיובי"

(4) אם $num > 100$ אזי

(4.1) הדפס את ההודעה "גדול"

.2

(1) קלוט מספר למשתנה num1

(2) קלוט מספר למשתנה num2

(3) קלוט מספר למשתנה total

(4) אם $num1 + num2 = total$ אזי

(4.1) הדפס את ההודעה "נכון!"

(5) אחרת

(5.1) הדפס את ההודעה "טעות!"

.3

(1) קלוט מספר למשתנה num1

(2) קלוט מספר למשתנה num2

(3) אם $num1 > num2$ אזי

(3.1) הדפס את הערך של num1

(3.2) הדפס את הערך של num2

(4) אחרת

(4.1) הדפס את הערך של num2

(4.2) הדפס את הערך של num1

.13

.1

(1) קלוט מספר למשתנה x

(2) אם $x \% 2 = 0$ וגם $x \% 3 = 0$ וגם $x \% 5 = 0$ אזי

(2.1) הדפס את ההודעה "מתחלק"

.2

(1) קלוט מספר למשתנה num1

(2) קלוט מספר למשתנה num2

(3) קלוט מספר למשתנה num3

(4) אם $num1 <> num2$ או $num1 <> num3$ או $num2 <> num3$ אזי

(4.1) הדפס את ההודעה "שונים"

.3

(1) קלוט מספר למשתנה a

(2) קלוט מספר למשתנה b

(3) קלוט מספר למשתנה c

(4) אם $a^2 = b^2 + c^2$ או $b^2 = a^2 + c^2$ או $c^2 = a^2 + b^2$ אזי

(4.1) הדפס את ההודעה "מתאים"

.14

.1

(1) קלוט מספר למשתנה x

(2) אם $x \% 2 = 0$ אזי

(2.1) אם $x \% 3 = 0$ אזי

(2.1.1) אם $x \% 5 = 0$ אזי

(2.1.1.1) הדפס את ההודעה "מתחלק"

.15

.1

(1) $mult \leftarrow 1$

(2) בצע 5 פעמים

(2.1) קלוט מספר למשתנה num

(2.2) $mult \leftarrow mult * num$

(3) הדפס את הערך של $mult$

.2

(1) $count \leftarrow 0$

(2) בצע 20 פעמים

(2.1) קלוט מספר למשתנה num

(2.2) אם $num < 0$ אזי

(2.2.1) $count \leftarrow count + 1$

(3) הדפס את הערך של $count$

.3

(1) $max \leftarrow 0$

(2) בצע 10 פעמים

(2.1) קלוט מספר למשתנה num

(2.2) אם $num > max$ אזי

(2.2.1) $max \leftarrow num$

(3) הדפס את הערך של max

.1

$$k \leftarrow 3 \quad (1)$$

(2) קלוט מספר למשתנה n

(3) בצע n פעמים

(3.1) הדפס את הערך של k

$$k \leftarrow k + 3 \quad (3.2)$$

.2

$$\text{sum} \leftarrow 0 \quad (1)$$

(2) קלוט מספר למשתנה n

$$i \leftarrow 1 \quad (3)$$

(4) בצע n פעמים

$$\text{sum} \leftarrow \text{sum} + i \quad (4.1)$$

$$i \leftarrow i + 1 \quad (4.2)$$

(5) הדפס את הערך של sum

.3

$$\text{mult} \leftarrow 1 \quad (1)$$

(2) קלוט מספר למשתנה n

$$i \leftarrow 2 \quad (3)$$

(4) בצע n פעמים

$$\text{mult} \leftarrow \text{mult} * i \quad (4.1)$$

$$i \leftarrow i + 2 \quad (4.2)$$

(5) הדפס את הערך של mult

.17

.1

(1) קלוט מספר למשתנה n

(2) עבור k מ-1 עד n בצע

(2.1) $a \leftarrow k * 3$

(2.2) הדפס את הערך של a

.2

(1) $sum \leftarrow 0$

(2) קלוט מספר למשתנה n

(3) עבור i מ-1 עד n בצע

(3.1) $sum \leftarrow sum + i$

(4) הדפס את הערך של sum

.3

(1) $mult \leftarrow 1$

(2) קלוט מספר למשתנה n

(3) עבור i מ-1 עד n בצע

(3.1) $mult \leftarrow mult * i * 2$

(4) הדפס את הערך של mult

.18

.1

(1) $fact \leftarrow 1$

(2) קלוט מספר למשתנה n

(3) עבור i מ-2 עד n בצע

(3.1) $fact \leftarrow fact * i$

(4) הדפס את הערך של fact

.2

sum \leftarrow 0 (1)

(2) קלוט מספר למשתנה n

(3) עבור i מ-1 עד n בצע

sum \leftarrow sum + nⁱ (3.1)

(4) הדפס את הערך של sum

.3

(1) קלוט מספר למשתנה n

(2) עבור i מ-0 עד n בצע

a \leftarrow 1 / (n + i) (2.1)

(2.2) הדפס את הערך של a

.19

.1

(1) קלוט מספר למשתנה from

(2) קלוט מספר למשתנה to

(3) כל עוד to \geq from בצע

(3.1) הדפס את הערך של to

(3.2) to \leftarrow to - 1

.2

(1) קלוט מספר למשתנה a

(2) קלוט מספר למשתנה b

power \leftarrow 1 (3)

(4) כל עוד b > 0 בצע

power \leftarrow power * a (4.1)

b \leftarrow b - 1 (4.2)

(5) הדפס את הערך של power

.3

- (1) קלוט מספר למשתנה num
- (2) $\text{prime} \leftarrow 1$
- (3) $i \leftarrow 2$
- (4) כל עוד $i \leq \sqrt{\text{num}}$ וגם $\text{prime} = 1$ בצע
 - (4.1) אם $\text{num} \% i = 0$ אזי
 - (4.1.1) $\text{prime} \leftarrow 0$
 - (4.2) $i \leftarrow i + 1$
 - (5) הדפס את הערך של prime

.20

.1

- (1) בצע
 - (1.1) קלוט מספר למשתנה grade1
 - (1.2) קלוט מספר למשתנה grade2
 - (2) עד ש $\text{grade1} \leq 100$ וגם $\text{grade1} \geq 0$ וגם $\text{grade2} \leq 100$ וגם $\text{grade2} \geq 0$
 - (3) $\text{average} \leftarrow (\text{grade1} + \text{grade2}) / 2$
 - (4) הדפס את הערך של average

.2

- (1) קלוט מספר למשתנה num1
- (2) קלוט מספר למשתנה num2
- (3) $\text{sum} \leftarrow \sqrt{\text{num1}} + \sqrt{\text{num2}}$
- (4) בצע
 - (4.1) הדפס את ההודעה "מהו סכום שורשי המספרים?"
 - (4.2) קלוט מספר למשתנה a
 - (5) עד ש $a = \text{sum}$

.3

(1) קלוט מספר למשתנה num

(2) $sum \leftarrow 0$

(3) בצע

(3.1) $sum \leftarrow sum + num \% 10$

(3.2) $num \leftarrow num - num \% 10$

(3.3) $num \leftarrow num / 10$

(4) עד ש $num \leq 0$

(5) הדפס את הערך של sum

.21

.1

(1) קלוט מספר למשתנה numbers(1)

(2) עבור i מ-2 עד 20 בצע

(2.1) $numbers(i) \leftarrow i + numbers(i - 1)$

.2

(1) $numbers(1) \leftarrow 0$

(2) $numbers(2) \leftarrow 1$

(3) עבור i מ-3 עד 15 בצע

(3.1) $numbers(i) \leftarrow numbers(i - 1) + numbers(i - 2)$

.3

(1) עבור i מ-1 עד 10 בצע

(1.1) קלוט מספר למשתנה array(i)

(1.2) $array(i) \leftarrow array(i) * i$

(2) עבור i מ-1 עד 10 בצע

(2.1) הדפס את הערך של $array(11 - i)$

פרק 2

תרשימי זרימה

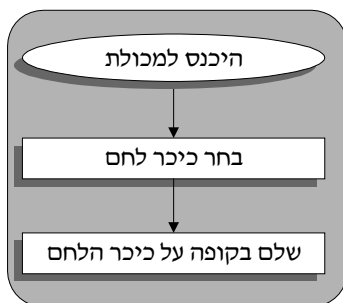
שאלות

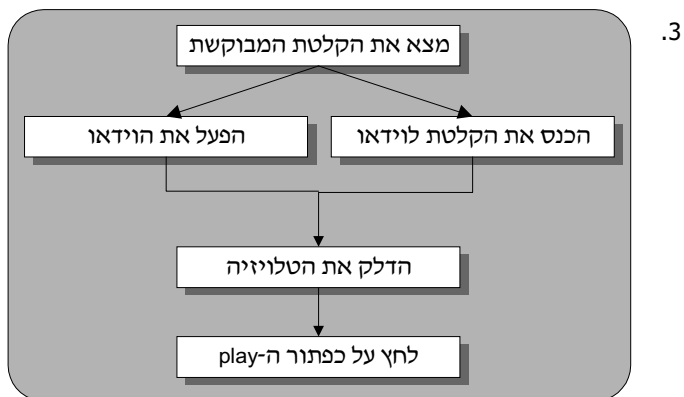
תרשימי זרימה פשוטים

תרשימי זרימה המיועדים לאדם

1. שרטט תרשימי זרימה לאלגוריתמים המופיעים בתרגיל 4 שבפרק 1.

2. ציין את השגיאה בכל אחד מתרשימי הזרימה שלפניך:





ייצוג פעולות אלגוריתמיות

פעולת הקלט

3. אחדות מהוראות הקלט שלפניך שגויות. סמן אותן והסבר מהן הטעויות.

<p>קלט a,b,c, d,e,f</p> <hr/> <p>4</p>	<p>קלט abcdef</p> <hr/> <p>3</p>	<p>קלט x y</p> <hr/> <p>2</p>	<p>קלט rem</p> <hr/> <p>1</p>
<p>קלט width</p> <hr/> <p>8</p>	<p>קלט len + 4</p> <hr/> <p>7</p>	<p>num</p> <hr/> <p>6</p>	<p>קלט age = 12</p> <hr/> <p>5</p>

פעולת הפלט

4. אחדות מהוראות הפלט שלפניך שגויות. סמן אותן והסבר מהן הטעויות. עבור הוראות הפלט הנכונות, ציין מה הן פולטות.

<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט a " " c</div> <hr style="width: 100%;"/> <p style="text-align: center;">4</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט 'hey'</div> <hr style="width: 100%;"/> <p style="text-align: center;">3</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט "age"</div> <hr style="width: 100%;"/> <p style="text-align: center;">2</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט age</div> <hr style="width: 100%;"/> <p style="text-align: center;">1</p>
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט a, "b", c</div> <hr style="width: 100%;"/> <p style="text-align: center;">8</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט "a", "b", "c"</div> <hr style="width: 100%;"/> <p style="text-align: center;">7</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט city</div> <hr style="width: 100%;"/> <p style="text-align: center;">6</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">פלט enema</div> <hr style="width: 100%;"/> <p style="text-align: center;">5</p>

פעולת ההשמה

5. ייצג כל אחת מההוראות הבאות על ידי פעולה אלגוריתמית מתאימה (אין קשר בין כל הוראה להוראה):

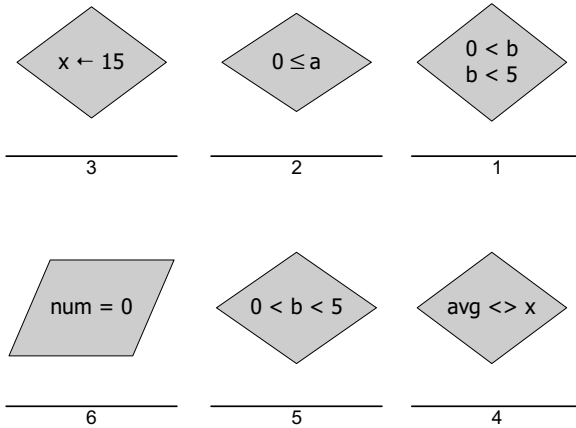
1. פלוט את המלל "Welcome"
2. הצב במשתנה area את שטח המלבן שאורכי צלעותיו מיוצגים על ידי ערכי המשתנים width ו-length
3. קלוט ערך לתוך המשתנה name
4. הצב במשתנה average את ממוצע ערכי המשתנים a, b ו-c
5. קלוט ערכים לתוך המשתנים day ו-month
6. הצב במשתנה perimeter את היקף המלבן שאורכי צלעותיו מיוצגים על ידי ערכי המשתנים width ו-length
7. פלוט את ערכי המשתנים lowest ו-highest
8. העלה בשלישית את הערך שבמשתנה power
9. פלוט את ערכי המשתנים first ו-last, כשביניהם המלל "-->"
10. הצב במשתנה area את שטח המעגל בעל הרדיוס radius

תרשימי זרימה המיועדים למחשב

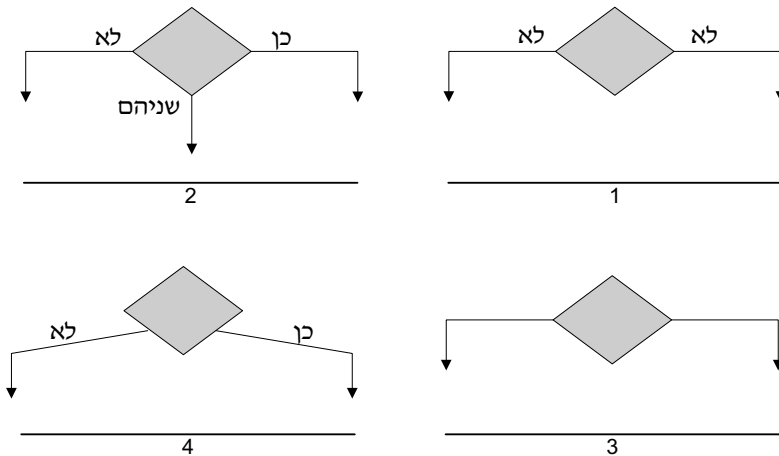
6. שרטט תרשימי זרימה לאלגוריתמים המופיעים בתרגיל 9 שבפרק 1.
 7. שרטט תרשימי זרימה לאלגוריתמים המופיעים בתרגיל 10 שבפרק 1.

תרשימי זרימה בעלי תנאים

8. אחדות מהוראות התנאי שלפניך שגויות. סמן אותן והסבר מהן הטעויות.



9. מצא את הטעויות באופני הוצאת החיצים מהוראות התנאי הבאות:



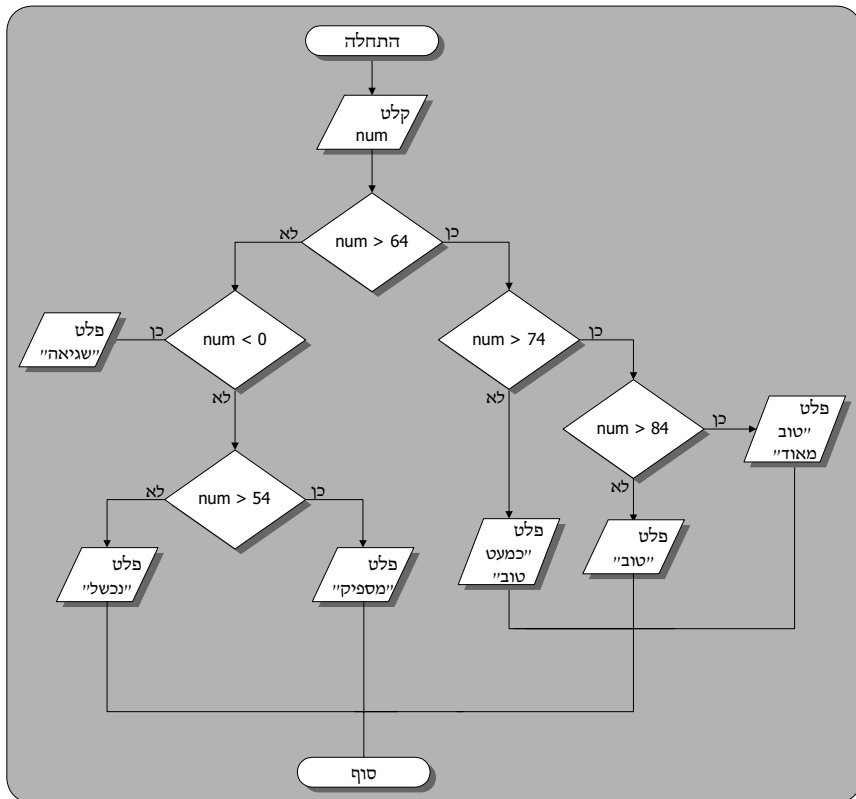
10. שרטט תרשימי זרימה לאלגוריתמים המופיעים בתרגיל 12 שבפרק 1.

הוראות תנאי מקוננות

11. שרטט תרשימי זרימה לאלגוריתמים הבאים:

- אלגוריתם הקולט שלושה מספרים מהמשתמש ופולט **שוים** אם הם שווים זה לזה.
 - אלגוריתם הקולט שני מספרים מהמשתמש ופולט אותם על המסך על-פי גודלם (המספר הגדול ייפלט ראשון, והקטן – שני). אם המספרים זהים, הם לא יוצגו על המסך, ותיפלט הודעת שגיאה.
 - אלגוריתם הקולט מספר מהמשתמש ופולט **מתחלק** אם הוא מתחלק ב-30 ללא שארית, על ידי שימוש במספרים 2, 3 ו-5 בלבד.
12. התבונן בתרשים הזרימה הבא וציין מה יהיה הפלט עבור כל אחד מהערכים הבאים של המשתנה num:

ערך num	הפלט המתקבל
-4	
44	
62	
88	
80	
65	
74	
55	



תרשימי זרימה בעלי לולאות

13. שרטט תרשים זרימה הפולט שלום n פעמים, כאשר n הוא מספר טבעי שמזין המשתמש. עשה זאת פעמים: פעם אחת תוך שימוש במשתנה עזר ופעם שנייה בלעדיו.

14. שרטט תרשימי זרימה לאלגוריתמים הבאים:

1. אלגוריתם הפולט את המספרים השלמים בין 1 ל- n (כולל), כאשר n הוא מספר טבעי שמזין המשתמש.

2. אלגוריתם הפולט את סכום המספרים השלמים בין 1 ל- n (כולל), כאשר n הוא מספר טבעי שמזין המשתמש.

3. אלגוריתם הפולט את סכום המספרים האי-זוגיים בין 1 ל- n (כולל), כאשר n הוא מספר טבעי שמזין המשתמש.

15. שרטט תרשימי זרימה לאלגוריתמים הבאים:

1. אלגוריתם הקולט מהמשתמש שני מספרים טבעיים, a ו- b , ופולט את החזקה שלהם (a^b) ללא שימוש באופרטור החזקה (^).

2. אלגוריתם הקולט מהמשתמש שני מספרים טבעיים ופולט את מכפלתם ללא שימוש באופרטור הכפל (*).

3. אלגוריתם הקולט מהמשתמש שני מספרים טבעיים ופולט את מנתם ללא שימוש באופרטור החילוק (/). הנח כי המספר הראשון גדול מהמספר השני והוא מתחלק בו ללא שארית.


16. שרטט תרשימי זרימה לאלגוריתמים הבאים:

1. אלגוריתם המציג שורה בת n כוכביות, כאשר n הוא מספר טבעי שמזין המשתמש. למשל, אם הוזן המספר 7, תוצג השורה הבאה:

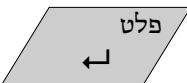
2. אלגוריתם המציג מלבן מלא מכוכביות, ברוחב w ובגובה h . הגדלים w ו- h הם מספרים טבעיים שמזין המשתמש. למשל, אם הוזנו הערכים 6 ו-3 בהתאמה, יוצג המלבן הבא:

שים לב!

בתרגיל זה ובתרגיל הבא עליך לפלוט את הכוכביות בשורות שונות. השתמש בהוראת הפלט הבאה כדי לרדת לשורה חדשה:



פלט

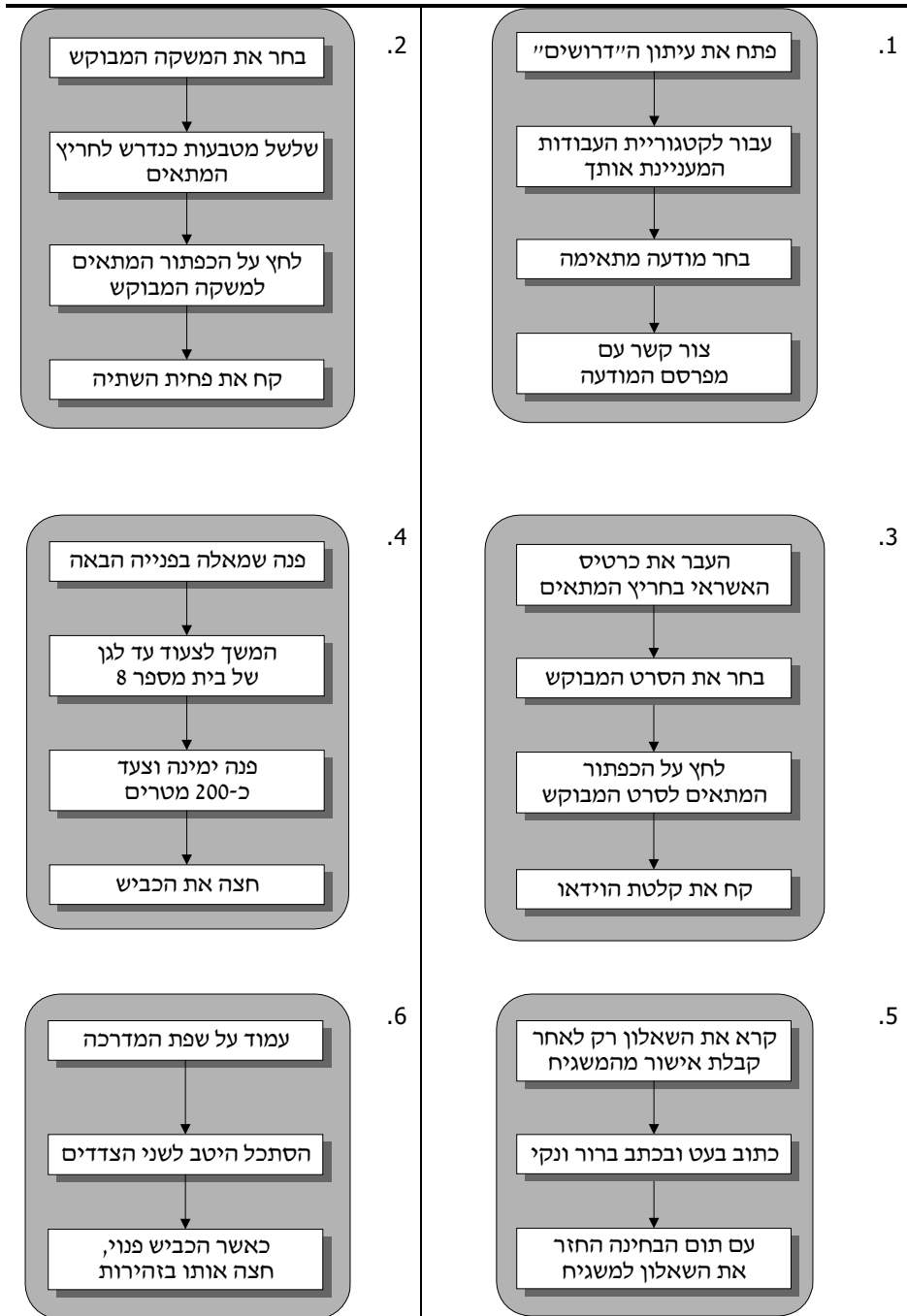


3. אלגוריתם המציג משולש ישר-זווית הבנוי מכוכביות, בגובה h , כאשר h הוא מספר טבעי שמזין המשתמש. למשל, אם הוזן הערך 5, יוצג המשולש הבא:

*
**

תשובות

.1



.2

1. יש לקשר בין הריבועים על ידי חיצים ולא על ידי קווים
2. על המטלות המיועדות לאדם להופיע במלבנים ולא באליפסות
3. אין להוציא שני חיצים מאותו מלבן

.3

1. x על הוראת הקלט להופיע במקבילית ולא במעוין	5. x בהוראת קלט יכולים להופיע שמות משתנים בלבד
2. x יש להפריד את שמות המשתנים אליהם נקלטים ערכים בפסיקים	6. x על המילה "קלט" להופיע בראש המקבילית
3. ✓	7. x בהוראת קלט יכולים להופיע שמות משתנים בלבד
4. ✓	8. x על הוראת קלט להיכתב במקבילית ולא במלבן

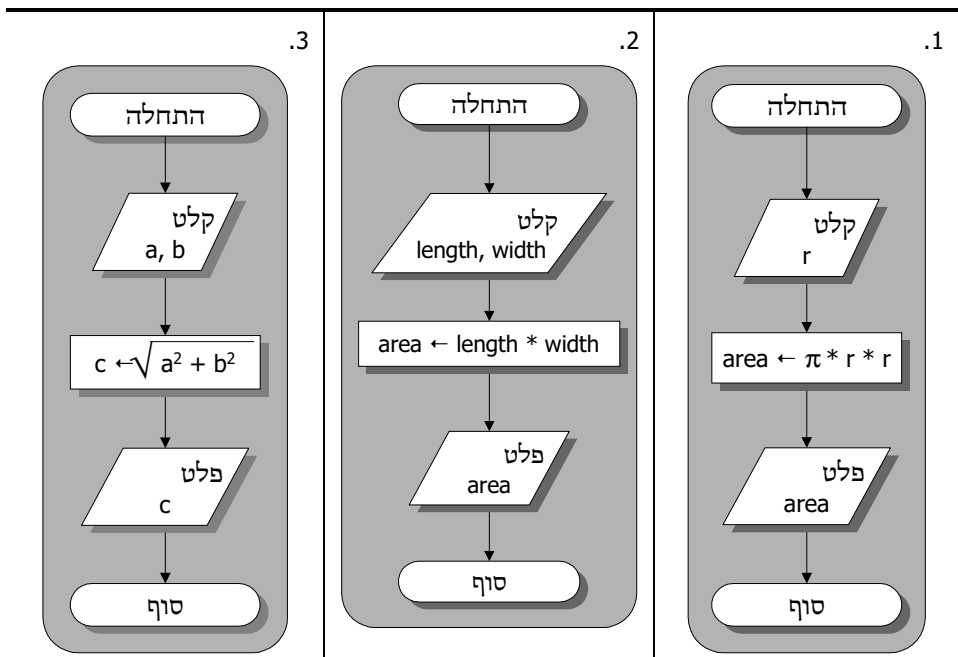
.4

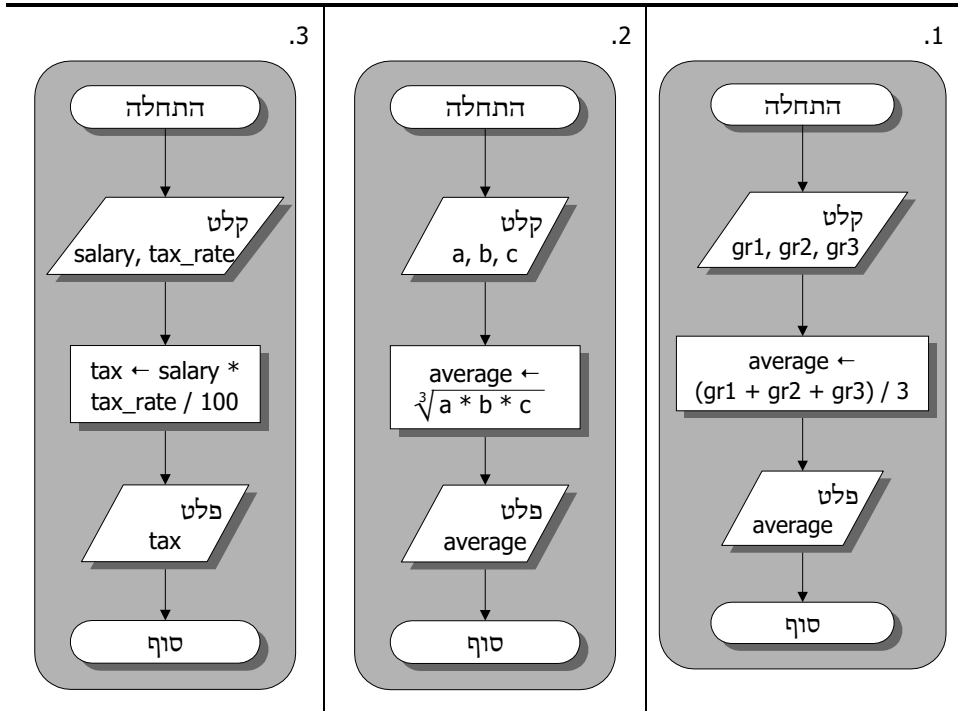
1. ✓ ההוראה פולטת את ערך המשתנה age	5. ✓ ההוראה פולטת את ערך המשתנה enema
2. ✓ ההוראה פולטת את המלל "age"	6. x מקבילית הפלט הפוכה
3. x על המלל הנועד לפליטה להיתחם בגרשיים כפולים (") ולא יחידים (')	7. ✓ ההוראה פולטת את המלל "abc"
4. x יש להפריד את המשתנים והמלל הנועדים לפליטה בפסיקים	8. ✓ ההוראה פולטת את ערכי המשתנים a ו-c, וביניהם את המלל "b"

.5

area ← length * width	.2	פלט "Welcome"	.1
average ← (a + b + c) / 3	.4	קלט name	.3
perimeter ← 2 * (width + length)	.6	קלט day, month	.5
power ← power ^ 3	.8	פלט lowest, highest	.7
area ← π * radius * radius	.10	פלט first, "-->", last	.9

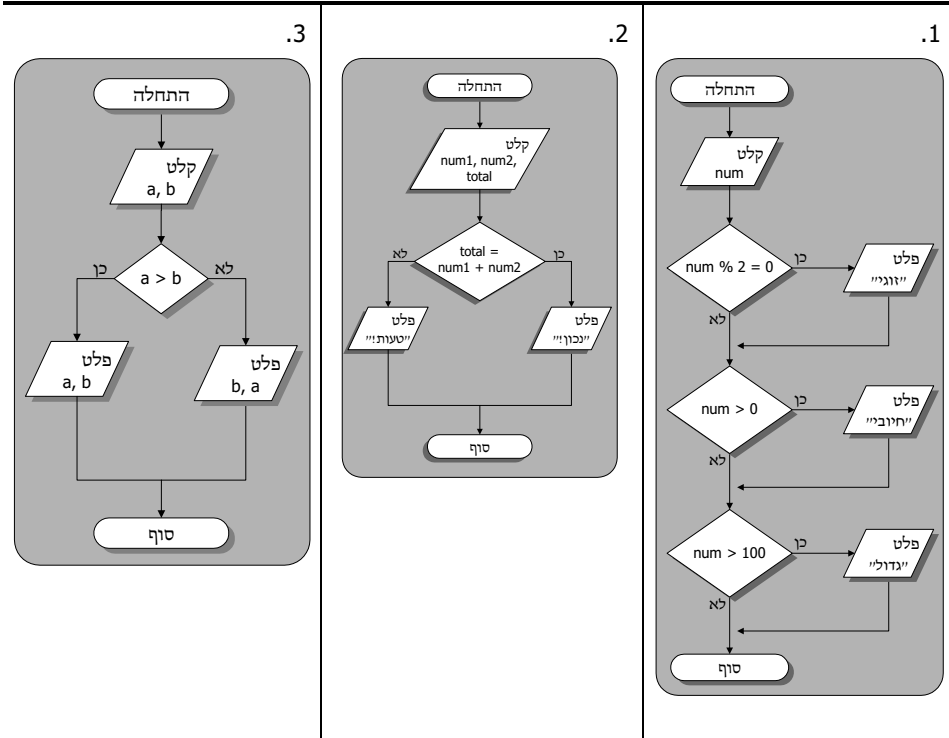
.6

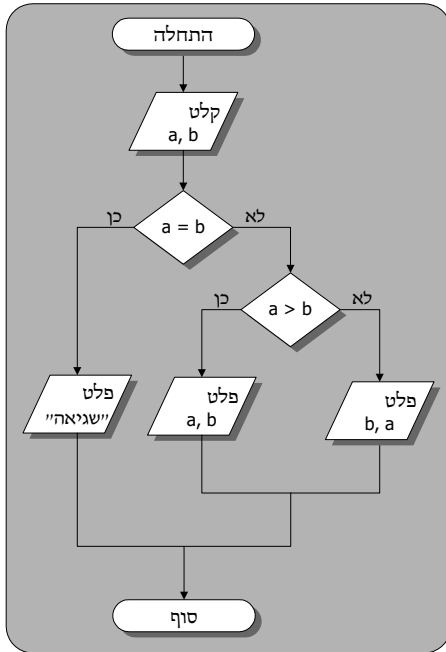
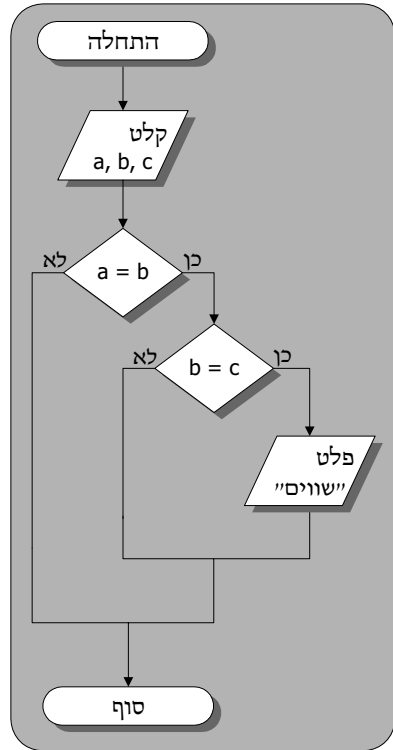


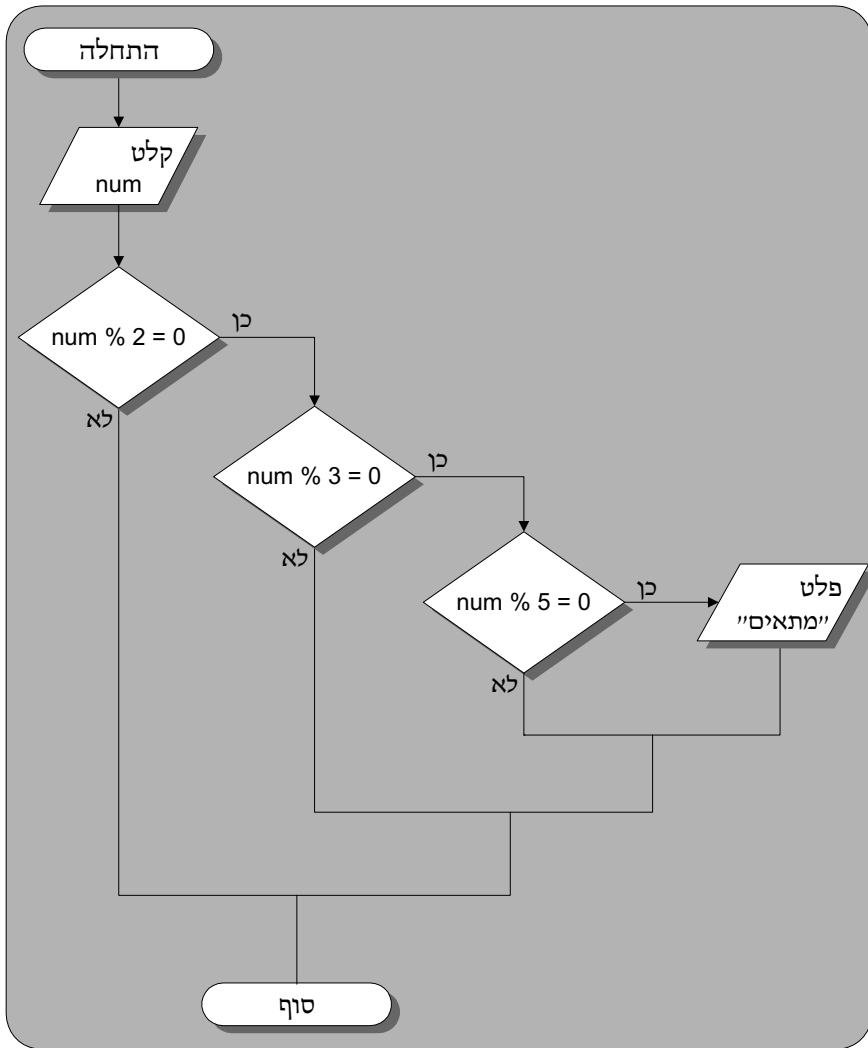


<p>4. ✓</p>	<p>1. x אין לבדוק שני תנאים בהוראת תנאי אחת</p>
<p>5. x אין לשלב שני אופרטורי יחס בהוראת תנאי אחת</p>	<p>2. ✓</p>
<p>6. x על הוראת התנאי להופיע במעוין</p>	<p>3. x החץ אינו יכול להופיע בהוראת התנאי</p>

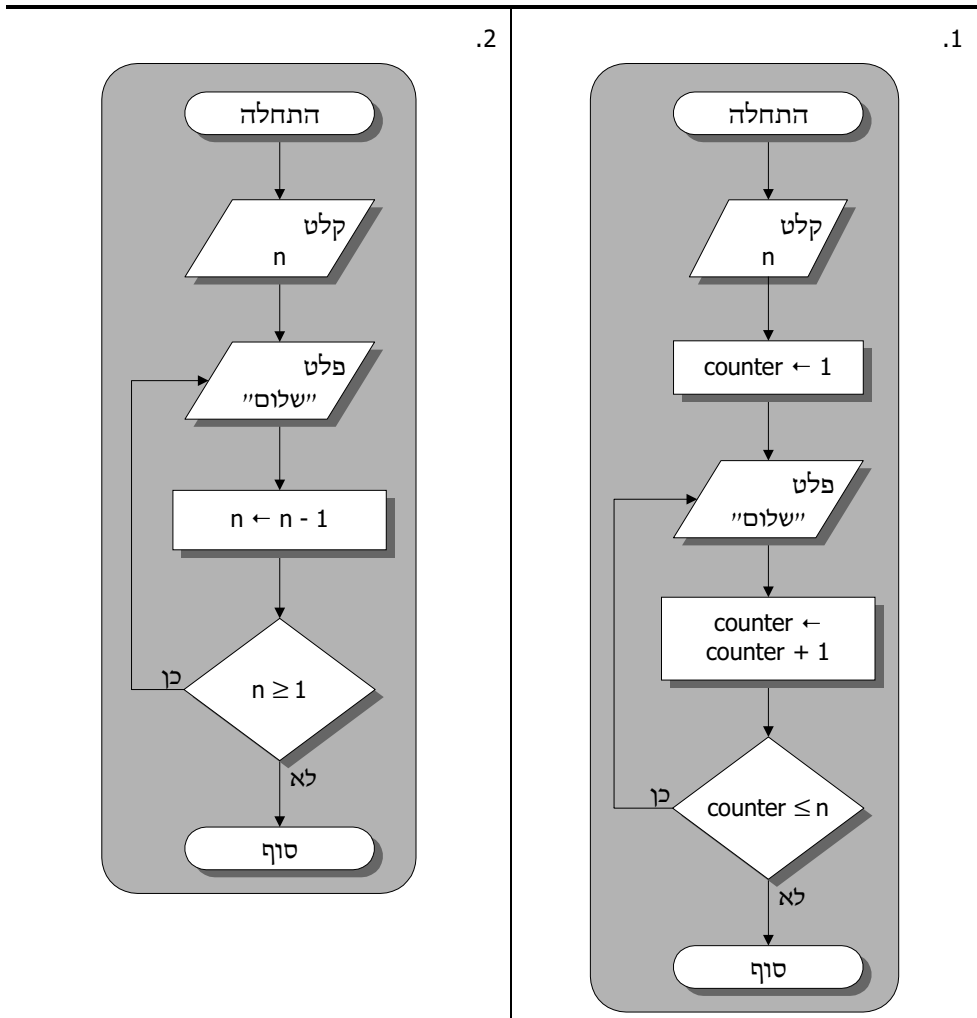
<p>3. יש לציין מעל כל חץ כן או לא</p>	<p>1. אין להוציא שני חיצי לא מאותה הוראת תנאי</p>
<p>4. על חיצי הוראת התנאי לצאת מקודקודיה</p>	<p>2. אין להוציא חץ שאינו כן או לא מהוראת תנאי</p>

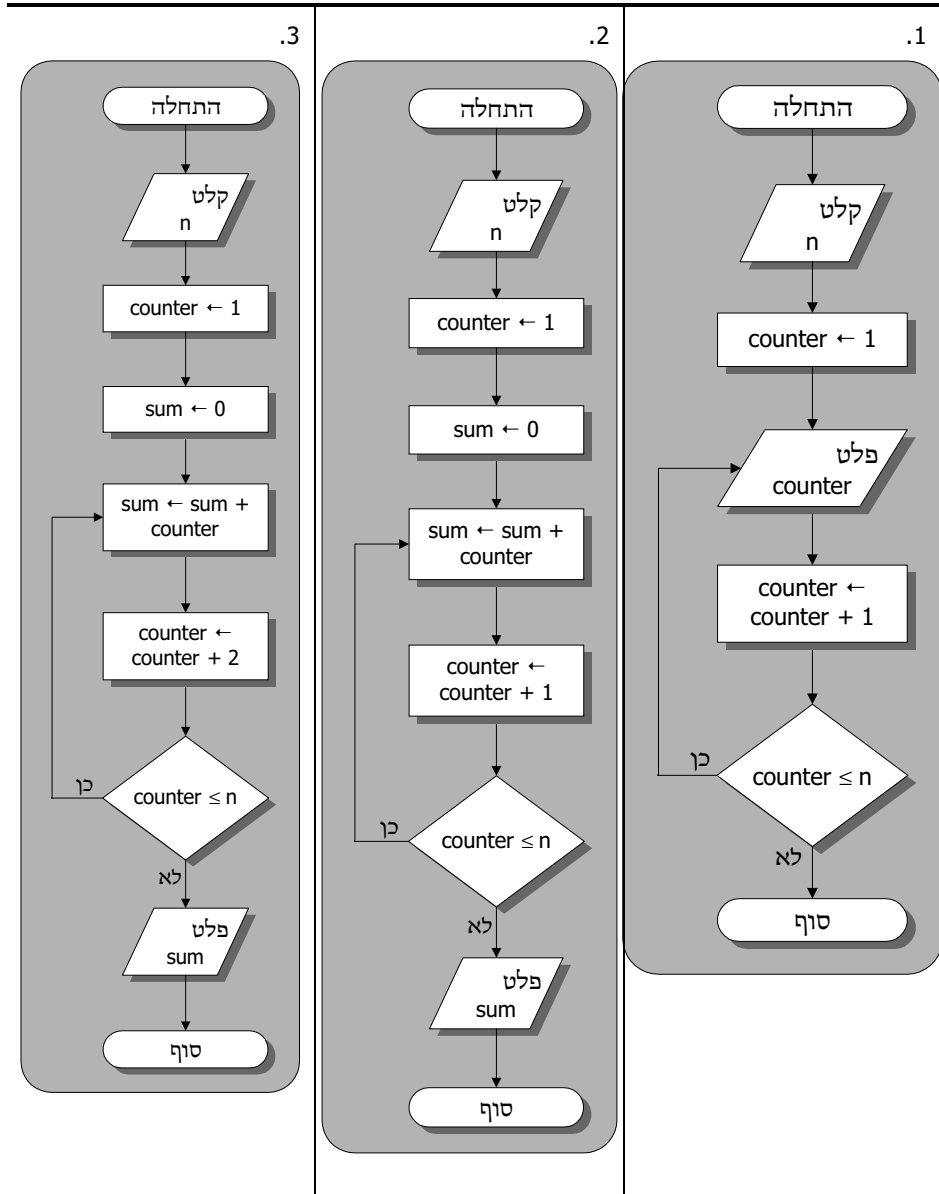


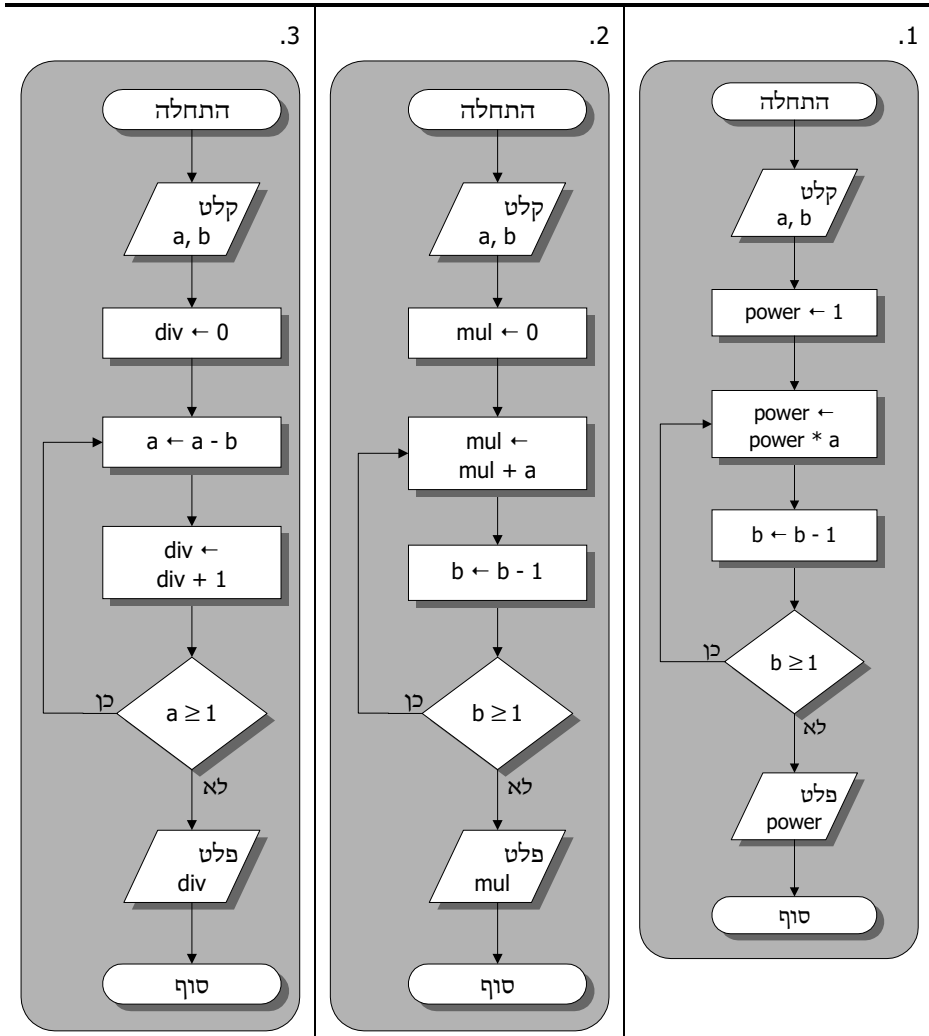


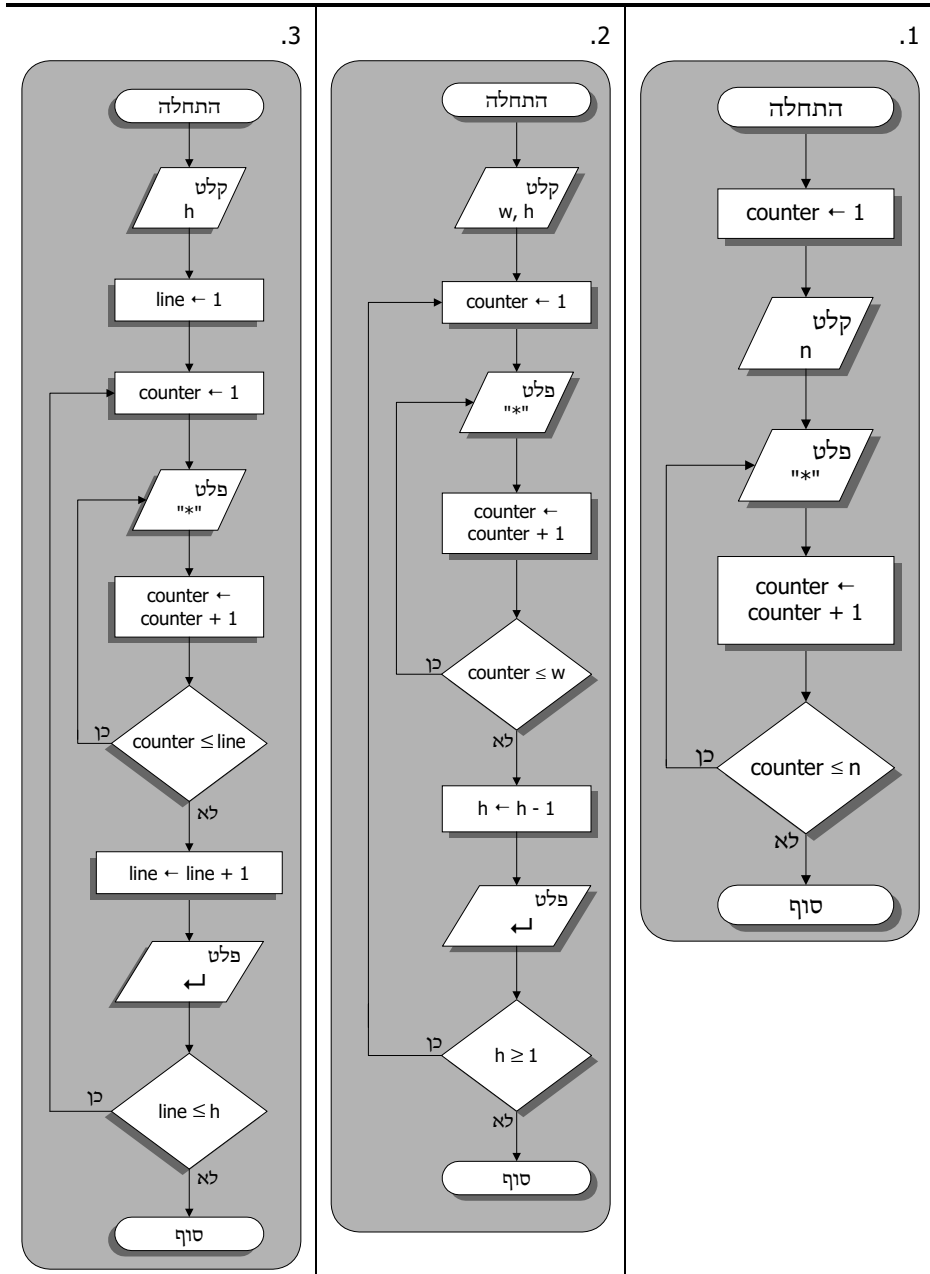


הפלט המתקבל	ערך num	הפלט המתקבל	ערך num
טוב	80	שגיאה	-4
כמעט טוב	65	נכשל	44
כמעט טוב	74	מספיק	62
מספיק	55	טוב מאוד	88









פרק 3

כלי הפיתוח של השפה

שאלות

סוגי קבצים

1. מלא את התרשים שלפניך, בהתאם לקבצים הבאים לידי ביטוי בעת יצירת קובץ הרצה.

שם הקובץ	תוכן הקובץ	סיומת נפוצה
<input type="text"/>		
↓		
<input type="text"/>		
↓		
<input type="text"/>		
↓		
<input type="text"/>		

יצירת הקבצים

2. הקף בעיגול את המשפטים הנכונים :

1. המהדר יוצר קובץ הרצה מקובץ המקור.
 2. בין היתר, המהדר יוצר קובץ המכיל הוראות אסמבלי, השקולות להוראות שהופיעו בקובץ המקור.
 3. בעת ביצוע אופטימיזציה פועל המהדר להקטנת קובץ האסמבלי.
 4. תוכנת הקשר מאחדת את קבצי האובייקטים לקובץ הרצה.
3. לפניך מספר פעולות המבוצעות על ידי המתכנת ועל ידי כלי הפיתוח של השפה. ציין את הסדר בו הן מתבצעות.

1. יצירת קובץ אובייקטים, שהוראותיו שקולות להוראות קובץ האסמבלי
2. ביצוע אופטימיזציה
3. כתיבת קוד מקור בתוכנת עריכה
4. איחוד קבצי האובייקטים לקובץ הרצה
5. ביצוע פעולות קדם-מעבד
6. יצירת קובץ אסמבלי, שהוראותיו שקולות להוראות קובץ המקור

יצירת קובץ ההרצה

4. הקף בעיגול את המשפטים הנכונים :

1. לרוב משפיע ריווח התוכנית על אופן ביצועה.
2. כמו המהדר, גם תוכנת הקשר עלולה להיתקל בבעיות בזמן פעולתה.
3. קובץ המקור צריך להיכתב בפורמט הכולל גופנים וסימנים גרפיים דומים.
4. בעת הפעלת המהדר משורת הפקודה, הדגלים מורים לו כיצד לפעול וכיצד להפעיל את תוכנת הקשר.

5. לפניך מספר תוכניות. הקלד אותן בעורך שברשותך, הדר, קשר והרץ אותן.

1.

```
#include <stdio.h>
void main()
{
    int x, y, area;

    printf("Enter width and height of rectangle: ");
    scanf("%d %d", &x, &y);
    area = x * y;
    printf("Area of rectangle: %d \n", area);
}
```

2.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char ch;

    printf("Please press a letter: ");
    ch = getch();
    putchar(ch);
    printf("\n3 Following letters: ");
    printf("%c %c %c \n", ch + 1, ch + 2, ch + 3);
}
```

3.

```
#include <stdio.h>
void main()
{
    int i, j, num;

    printf("Please enter a positive number: ");
    scanf("%d", &num);

    for (i = 0 ; i < num ; i++) {
        for (j = 0 ; j <= i ; j++)
            printf("*");
        printf("\n");
    }
}
```

עבודה עם המהדר

הודעות שגיאה / הודעות אזהרה

6. הקף בעיגול את המשפטים הנכונים :

1. הודעות שגיאה מתעוררות כתוצאה מטעויות תחביריות.
2. הודעות שגיאה מציגות תמיד את השורה המדויקת בה נעשתה הטעות.
3. טעות בודדת עלולה לגרום לפליטת מספר הודעות שגיאה ואזהרה.
4. יש להתעלם מהודעות אזהרה שפולט המהדר.

התנסות עם הודעות המהדר

7. לפניך תוכנית נכונה בשפת C, הקולטת מספר מהמשתמש ומציגה את סכום המספרים הטבעיים עד אליו :

```
1 #include <stdio.h>
2 void main()
3 {
4     int sum = 0, i, n;
5
6     printf("This program calculates the sum ");
7     printf("of natural numbers from 1 to n. \n");
8     printf("Please enter n: ");
9     scanf("%d", &n);
10
11     for (i = 1 ; i <= n ; i++)
12         sum += i;
13
14     printf("Sum is: %d \n", sum);
15 }
16
```

בצע את הפעולות הבאות (כל אחת בנפרד) והתרשם מההודעה המתקבלת :

- ❖ מחק את הסוגריים הסוגרים בסוף שורה 6.
- ❖ מחק את הסוגריים המסולסלים הפותחים בשורה 3.
- ❖ מחק את אחד הפסיקים בשורה 4.

תשובות

.1

סיימת נפוצה	תוכן הקובץ	שם הקובץ
.c, .h	הוראות בשפת C (קוד מקור)	קובץ מקור
		⇓
.asm	הוראות קובץ המקור – בשפת סף	קובץ אסמבלי
		⇓
.obj, .o	הוראות קובץ האסמבלי – בשפת מכונה	קובץ אובייקטים
		⇓
.exe	הוראות קובץ האובייקטים + קבצי אובייקטים נוספים	קובץ הרצה

.2

.3 ✓	.1 x
.4 ✓	.2 ✓

.3

.3
.5
.6
.2
.1
.4

		.4
.3 x		.1 x
.4 ✓		.2 ✓
		.6
.3 ✓		.1 ✓
.4 x		.2 x

פרק 4

אבני היסוד של השפה

שאלות

תוכנית שלד

1. הקף בעיגול את המשפטים הנכונים :

1. תוכנית שלד היא התוכנית הקצרה ביותר שניתן לכתוב בשפת תכנות כלשהי.
2. תוכנית השלד מבצעת פעולת פלט אחת כאשר היא מורצת.
3. הוראות תוכנית השלד תופענה בכל תוכנית שנכתוב בעתיד.
4. הוראותיה של תוכנית בשפת C מקבילות בדיוק להוראות האלגוריתם המתאים לה.
5. הפונקציה הראשית מכילה הוראות המתבצעות בכל פעם שמופעל קובץ ההרצה.
6. בתוך הפונקציה הראשית חייבת להופיע הוראה אחת לפחות.

תווים לבנים

2. אחדות מתוכניות השלד שלפניך שגויות. סמן אותן והסבר מהן הטעויות.

```
1.  
void main{  
(  
  
)
```

2.

```
void main()  
    {  
    }
```

3.

```
void main  
{  
}
```

4.

```
void main() { }
```

5.

```
void mainfunction()  
{  
}
```

6.

```
void main()  
}  
  
{
```

7.

```
VOID MAIN()  
{  
}
```

8.

```
void ma in()  
}  
  
{
```

פלט בסיסי

3. אחדות מהוראות הפלט שלפניך שגויות. סמן אותן והסבר מהן הטעויות.

1. `printf("Programming in C");`
2. `printf(hello, world!);`
3. `printf("")`
4. `printf("printf(hello, world!);");`
5. `printf "We have not yet begun";`
6. `PrintF("void main()");`
7. `printf(" we're leaving together ");`
8. `printf('Alert! Alert!');`

4. ציין את השגוי בכל אחת מהתוכניות שלפניך:

```
1.
void main()
{
    printf("A slumber did my spirit seal");
}
```

```
2.
include <stdio.h>
void main()
{
    printf("A slumber did my spirit seal");
}
```

```
3.
#include <stdio.h>
void main()
{
}
printf("A slumber did my spirit seal");
```

```
4.
#include stdio.h
void main()
{
    printf("A slumber did my spirit seal");
}
```

ירידה לשורה חדשה

5. ציין מהו הפלט המתקבל מכל אחד מקטעי הקוד הבאים :

1.

```
printf("Far over the misty mountains cold \n");
printf("\nTo dungeons deep and caverns old \n");
printf("We must away ere break of day,");
printf("\n To find our long-forgotten gold.");
```

2.

```
printf("Traggic jam, \nwhen you're already");
printf(" late \n\na no smoking sign, \non ");
printf("your cigarette break \n\nIt's like te");
printf("n thousand spoons, \nwhen all you need ");
printf("is a knife.");
```

3.

```
printf("Let's all meet up \nin the year 2000");
printf(", \nwon't it be strange \nwhen we're ");
printf("all fully grown");
printf(".");
```

הערות

6. מצא את השגיאה בכל אחת מההערות שלהלן :

1.

```
void main() /* main part of the program */
```

2.

```
/*
 * This magnificent program
 * does absolutely nothing.
```

3.

```
/* this program was created by TNL
for more information, call 1-800-DRAGON
/* not between 14:00 and 16:00 ! */
thank you.
*/
```

4.

```
print/*display something*/f("Enter a number: ");
```

תווי בקרה

7. ציין מהו הפלט של כל אחת מהוראות הפלט הבאות:

1.

```
printf("Oh where the north wall \n\tis crumbling, \nLet us put  
mortar and brick, \nLet us stack limestone \n\t\non  
limestone, \nLaid down with a promise \n\t\nand lick.");
```

2.

```
printf("if she's a liar, \n\tI'm her lover \a\nif she's a  
priestess, \n\tI'm her cover \nif she's a lady, \n\tI'm her  
man \a\nif she's a man, \n\tI'll do what I can. \a");
```

3.

```
printf("\tA simple Child \a\nThat lightly \n\t\tdraws \n\t\tits  
breath \a\nAnd feels its life \a\n\tin every limb, \nWhat  
should it \n\tknow of \n\t\tdeath? \a");
```

8. כתוב תוכניות המציגות את הפלטים הבאים:

1.

```
Please enter a digit\letter\character:
```

2.

```
KNOCK KNOCK  
"Who's there?"  
"Who."  
"Who who?"  
"What are you, an owl?"
```

3.

```
KNOCK KNOCK  
"Who's there?"  
"Interrupting cow."  
"Interrupting c..."  
"Moooooo."
```

4.

```
KNOCK KNOCK  
"Who's there?"  
"Anita."  
"Anita who?"  
"Anita tissue... ah-choo!! Too late!"
```

מרכיבי השפה

9. כתוב לצד כל אחד ממרכיבי השפה הבאים את הספרה המתאימה לקבוצתו, בהתאם למקרא הבא:

ספרה מתאימה	רכיב שפה
1	מילות מפתח
2	מזהים
3	קבועים
4	אופרטורים
5	סימני פיסוק

1.	12.8	10.	-
2.	extern	11.	[]
3.	"abcde"	12.	max_height
4.	+	13.	'\t'
5.	default	14.	p
6.	x	15.	#
7.	-	16.	/
8.	unsigned	17.	;
9.	MyFunction	18.	765

תשובות

1.

.1 ✓	.4 ✗
.2 ✗	.5 ✓
.3 ✓	.6 ✗

2.

.1 ✗	על הסוגריים המופיעים לאחר שם הפונקציה main להיות סוגריים רגילים ועל הסוגריים הקובעים את גוף הפונקציה להיות סוגריים מסולסלים.
.2 ✓	
.3 ✗	לאחר שם הפונקציה הראשית יש להוסיף סוגריים.
.4 ✓	
.5 ✗	על הפונקציה הראשית להיקרא main.
.6 ✗	על הסוגריים הקובעים את גוף הפונקציה להיות פותחים וסוגרים, בהתאמה, ולא סוגרים ופותחים.
.7 ✗	את המילים void ו-main יש לרשום באותיות קטנות.
.8 ✗	אין להוסיף רווח בין תווי המילה main.

3.

.1 ✓	.5 ✗	יש לתחום את הפרמטר הנשלח לפונקציה בסוגריים
.2 ✗	.6 ✗	על שם הפונקציה printf להיכתב באותיות קטנות
.3 ✗	.7 ✓	יש לסיים את הקריאה לפונקציה בנקודה-פסיק
.4 ✓	.8 ✗	יש לתחום את המלל הנשלח לפונקציה בגרשיים

.4

.1	חסרה הוראת #include.
.2	חסרה סולמית (#) לפני המילה include.
.3	אין לבצע קריאה לפונקציה מחוץ לגוף הפונקציה.
.4	על שם הקובץ המוכלל להתחם בסוגריים משולשים (> ו- <).

.5

1.	Far over the misty mountains cold To dungeons deep and caverns old We must away ere break of day, To find our long-forgotten gold
2.	Traffic jam, when you're already late a no smoking sign, on your cigarette break It's like ten thousand spoons, when all you need is a knife.
3.	Let's all meet up in the year 2000, won't it be strange when we're all fully grown.

.6

.1	אין להוסיף רווח בין תווי תוחמי הערות.
.2	יש לסיים כל הערה ב- /*.
.3	אין לקנן הערות אחת בשנייה.
.4	אין לכתוב הערה באמצע שמה של הוראה.

-
1. Oh where the north wall
 is crumbling,
 Let us put mortar and brick,
 Let us stack limestone
 on limestone,
 Laid down with a promise
 and lick.
2. if she's a liar,
 I'm her lover ♪
 if she's a priestess,
 I'm her cover
 if she's a lady
 I'm her man ♪⁻
 if she's a man,
 I'll do what I can. ♪⁻
3. A simple Child ♪⁻
 That lightly
 draws
 its breath ♪⁻
 And feels its life ♪⁻
 in every limb,
 What should it
 know of
 death? ♪⁻
-

1.

```
#include <stdio.h>
void main()
{
    printf("Please enter a digit\\letter\\character:");
}
```

2.

```
#include <stdio.h>
void main()
{
    printf("KNOCK KNOCK\n\"Who's there?\"\n\"Who.\"\n\"Who
        who?\"\n\"What are you, an owl?\"");
}
```

3.

```
#include <stdio.h>
void main()
{
    printf("KNOCK KNOCK\n\"Who's there?\"\n\"Interrupting
        cow.\"\n\"Interrupting c...\"\n\"Moooooo.\"",);
}
```

4.

```
#include <stdio.h>
void main()
{
    printf("KNOCK KNOCK\n\"Who's there?\"\n\"Anita.\"\n\"Anita
        who?\"\n\"Anita tissue... ah-choo!! Too late!\"",);
}
```

1.	3	7.	4	13.	3
2.	1	8.	1	14.	2
3.	3	9.	2	15.	5
4.	4	10.	4	16.	4
5.	1	11.	5	17.	5
6.	2	12.	2	18.	3

פרק 5

משתנים

שאלות

מהו משתנה?

1. הקף בעיגול את המשפטים הנכונים:

1. השימוש במשתנים מאפשר קליטת ערכים מהמשתמש וביצוע חישובים שונים.
2. כל משתנה תופס שטח ניכר מזיכרון המחשב.
3. בשפת C סוג משתנה אחד ויחיד.
4. כל משתנה מיוצג על ידי שמו, אותו ניתן להרכיב מאותיות לועזיות, מספרות ומקו תחתי.
5. כל המשתנים בשפת C תופסים שטח זהה בזיכרון המחשב.
6. אין הגבלה על השמות שניתן להעניק למשתנים.

הצבת ערכים

2. לפניך מספר משפטי השמה. מצא מהו הערך המוצב במשתנה בכל אחד מהם. התייחס לערך המשתנה salary כ-100, לערך המשתנה base כ-4, לערך המשתנה max כ-5 ולערך המשתנה num כ-7.

- | | |
|-------------------------|--------------------------------|
| 1. value = 10 - 4; | 7. div = 9 / 4.5 / 2; |
| 2. tax = 0.19 * salary; | 8. avg = 4 % 2 + - 7; |
| 3. extra = 17 % 3; | 9. delta = 4 - -num * max; |
| 4. sqr = base * base; | 10. len = 5 / (4 % 3) - 7 + 5; |
| 5. min = -max + 1; | 11. num = num * num; |
| 6. length = 3 % 10 * 2; | 12. sum = sum - sum; |

3. מצא את הטעות בכל אחד ממשפטי ההשמה הבאים:

- | | |
|----------------------|-----------------------------|
| 1. x = 10 | 5. value ← 7; |
| 2. power = 4 ^ 2; | 6. x + y = 13; |
| 3. 9 = grade; | 7. ran := 100; |
| 4. pure = tot / 0.0; | 8. sqr_root = $\sqrt{25}$; |

סוגי משתנים

המשתנה השלם / המשתנה העשרוני

4. באיזה סוג משתנה (שלם או עשרוני) נבחר לייצג כל אחד מהערכים הבאים?

- | | |
|----------------------------|----|
| מספר השרים בממשלה | .1 |
| מחיר מוצר בסופרמרקט (בש"ח) | .2 |
| מהירות הסיבוב של כוכב לכת | .3 |
| מספר זוגות הנעליים שברשותך | .4 |
| מימדי בית | .5 |
| טמפרטורה (במעלות צלסיוס) | .6 |
| מספר השירים בתקליטור | .7 |
| משקל אלקטרון (בק"ג) | .8 |

דיוק בערכים שלמים ועשרוניים

5. מצא מהם ערכי המשתנים i ו-d לאחר ביצוע משפטי ההשמה הבאים. הנח כי המשתנה i הוא משתנה שלם (int) והמשתנה d הוא משתנה עשרוני (float).

- | | | | |
|----|------------------------------|-----|-------------------------------|
| 1. | <code>i = 5 * 7;</code> | 8. | <code>i = 0e0;</code> |
| 2. | <code>d = 8;</code> | 9. | <code>d = 123e2;</code> |
| 3. | <code>i = 27.0 / 9.0;</code> | 10. | <code>i = 0.9999;</code> |
| 4. | <code>i = 6 / 4;</code> | 11. | <code>i = 13.0 / 4.0;</code> |
| 5. | <code>d = 6 / 4.0;</code> | 12. | <code>d = 5 / 2;</code> |
| 6. | <code>d = 6 / 4;</code> | 13. | <code>i = 5 / 2.0;</code> |
| 7. | <code>d = 0e0;</code> | 14. | <code>d = 99e5 / 10e3;</code> |

הגדרת משתנים

6. כתוב שורות הגדרה מתאימות למשתנים הבאים:

- | | |
|----|---|
| .1 | <code>i</code> (שלם) |
| .2 | <code>age</code> (עשרוני) |
| .3 | <code>tax, salary</code> (עשרוניים) |
| .4 | <code>base, exp</code> (שלמים) |
| .5 | <code>grade3, grade2, grade1</code> (שלמים) |
| .6 | <code>avg3, avg2, avg1</code> (עשרוניים) |

7. מצא את הטעות בכל אחת משורות ההגדרה הבאות:

- | | |
|----|---------------------------------|
| 1. | <code>power;</code> |
| 2. | <code>num1 int;</code> |
| 3. | <code>float pow</code> |
| 4. | <code>int x y z;</code> |
| 5. | <code>float, i, j;</code> |
| 6. | <code>int a, b, float c;</code> |

8. מצא מהם ערכי המשתנים i ו-d בסוף כל אחת מהתוכניות הבאות:

```
1. void main()
{
    int i;
    float x;

    i = 10.5;
    x = 30.0;
    i = i % 6;
    i = i - 2;
    x = x / i + 3;
    i = i + 1;
    x = x / i;
}
```

```
2. void main()
{
    int i, x;

    x = 3;
    i = -x - -x;
    x = i / 3 * 4.8;
    i = -2 * 3 - i;
    i = x - i;
    x = -i + x;
    i = i + x;
}
```

פלט ערכי משתנים

9. מהו הפלט המתקבל מכל אחת מהוראות הפלט הבאות?

היעזר בנתונים הבאים:

שם המשתנה	סוג המשתנה	ערך המשתנה
num	שלם	11
speed	עשרוני	5.43
channel1	שלם	3
channel2	שלם	4
kids	עשרוני	2.5
vel	שלם	1

1. printf("%d trees flourish at night", num);
2. printf("Fighting ghosts on July %dth", 6);
3. printf("Neptune's orbital speed is %f km/s", speed);
4. printf("I mostly like channels %d and %d", channel1, channel2);
5. printf("%d pirates eating %f kids", 6, kids);
6. printf("%d m/s is %f mi/h and %f cm/s", vel, 2.237, 100.0);

10. בתרגיל זה ניווכח עד כמה חשובה ההתאמה בין תו ההמרה לפרמטר שלו.

❖ הרץ כל אחת מהוראות הפלט הבאות במחשבך וציין מהו הפלט המתקבל מכל אחת מהן.

❖ תקן כל הוראת פלט, כך שתפלוט מלל זהה לזה שפלטתה הוראת הפלט המתאימה לה בתרגיל הקודם.

1. `printf("%f trees flourish at night", 11);`
2. `printf("Fighting ghosts on July %dth", 6.0);`
3. `printf("Neptune's orbital speed is km/s", 5.43);`
4. `printf("I mostly like channels %d and %d");`

קליטת ערכים מהמשתמש

11. אחדות מהוראות הקלט שלפניך שגויות (הוראה שגויה היא הוראה שאינה מבצעת את תפקידה כיאות). סמן אותן והסבר מהן הטעויות. התייחס אל המשתנים i_2, i_1 ו- i_3 כמשתנים שלמים ואל המשתנים f_1, f_2 ו- f_3 כמשתנים עשרוניים.

1. `scanf("%d %d", &i1, &i2);`
2. `scanf("%f");`
3. `scanf("%f", &i3);`
4. `scanf("%d %d %f %f", &i1, &i2, &f1, &f2);`
5. `scanf("Enter a number (1-100): %d", &i1);`
6. `scanf("%d %f", i3, f3);`
7. `scanf("%f%f", &f1, &f2);`
8. `scanf("%l f % d", &f2);`

12. כתוב את התוכניות הבאות:

1. תוכנית הקולטת אורכי צלעות של מלבן ופולטת את שטחו ואת היקפו.
2. תוכנית הקולטת רדיוס של מעגל ופולטת את שטחו ואת היקפו.
3. תוכנית הקולטת את מהירותה הקבועה של מכונית (בקמ"ש) ואת זמן נסיעתה (בשעות) ופולטת את המרחק שעברה (במטרים).
4. תוכנית הקולטת את המסה של אדם בק"ג ופולטת אותה בפאונדים (1 ק"ג = 2.205 פאונד).

המשתנה התווי

13. לפניך מספר משפטי השמה, חלקם נכונים וחלקם שגויים. הקף את המשפטים השגויים וציין מהי הטעות בהם. עבור כל משפט נכון, ציין את ערך המשתנה התווי ch לאחר ביצועו.

1. `ch = '$';`
2. `ch = 'a + 5';`
3. `ch = 'a' + 6;`
4. `ch = 9 * 5;`
5. `ch = "3";`
6. `ch = '\\';`
7. `ch = '%d';`
8. `ch = '\';`
9. `ch = 100;`
10. `ch = '100';`

פלט תווים / קליטת תווים

14. כתוב את התוכניות הבאות:

1. תוכנית הקולטת תו מהמשתמש ופולטת את ארבעת התווים הבאים אחריו, על-פי טבלת ה-ASCII.
2. תוכנית הקולטת אות קטנה מהמשתמש ופולטת את האות הגדולה המתאימה לה.
3. תוכנית הקולטת מהמשתמש שתי אותיות לועזיות, הראשונה גדולה והשנייה קטנה, ופולטת מספר דו-ספרתי, שספרת האחדות שלו מתאימה לאות הקטנה וספרת העשרות שלו מתאימה לאות הגדולה. הספרות מתאימות לאותיות במונח שהאות 'a' (או 'A') מסמלת את הספרה 0, האות 'b' (או 'B') מסמלת את הספרה 1, האות 'c' (או 'C') מסמלת את הספרה 2, וכן הלאה - עד האות 'j' (או 'J'), המסמלת את הספרה 9. בתוכנית זו פלוט וקלוט תווים בלבד (לא מספרים).

הפונקציה getch / הפונקציה putchar

15. לפניך תוכנית האמורה לקלוט תו מהמשתמש ולפלוט אותו שלוש פעמים. התוכנית שגויה בשל היעדרה של שורה אחת. הוסף שורה זו לתוכנית.

```
#include <stdio.h>
void main()
{
    char ch;

    printf("Please press a key: ");
    ch = getch();

    putchar(ch);
    putchar(ch);
    putchar(ch);
}
```

16. כתוב את התוכניות שבתרגיל 14 תוך שימוש בפונקציה getch, כך שהתווים שיוזין המשתמש לא יוצגו על המסך.

פלט מורכב באמצעות printf

17. מצא מה הפלט המתקבל מכל אחת מהוראות ה-printf הבאות:

1. printf("%.2f", 1.2345);
2. printf("%.3f", 4.0)
3. printf("%.1f", 99.75);
4. printf("%.0f", 99.75);
5. printf("%6.2f", 123.45);
6. printf("%4.1f", 33333.0);
7. printf("%11f", 111.222);
8. printf("%5d", 76);

אתחול משתנים

18. אחדות משורות ההגדרה שלפניך שגויות. סמן אותן וכתוב אותן כך שיהיו נכונות מבחינה תחבירית:

1. float g = 9.78; g2 = 9.81;
2. int num = 12 * 3 - 6;
3. int 25 = age;
4. char letter, digit, dif = '9' - '0', DIF;
5. char ch1 = 'b', int num1 = 10;
6. int room = 1, int apart = 2;
7. float n1 = 4.0, n2 = n1, n3 = n1 * n2;
8. int f1 = f2, f2 = 0;

אופרטורים מקוצרים

אופרטורי השמה מקוצרים

19. מצא משפטי השמה מתאימים למשפטי ההשמה הבאים, תוך שימוש באופרטורי השמה מקוצרים:

- | | |
|---------------------------|------------------------------|
| 1. home = home + 10; | 5. kg = kg * (3 + mass); |
| 2. price = price * 1.205; | 6. min = min - 1; |
| 3. exp = exp + 1; | 7. col = col % (7 * row); |
| 4. vol = vol * vol * vol; | 8. month = month - tot + 10; |

אופרטורי העלאה והפחתה עצמית

20. אחדים ממשפטי ההשמה שלפניך שגויים. סמן אותם והסבר מהן הטעויות.

- | | |
|----------------------|-----------------|
| 1. distance + = 100; | 6. highest ++; |
| 2. temp++; | 7. pass++++; |
| 3. (faran + 5)--; | 8. weight*=100; |
| 4. a + b %= c; | 9. --avg--; |
| 5. eq =* 3.33; | 10. phys++math; |

ביטויים מורכבים

ערך מוחזר מאופרטורי העלאה והפחתה עצמית

21. לפניך מספר ביטויים בהם מעורבים שלושה משתנים שלמים (a, b ו-c), ערכים שלמים ואופרטורים מסוגים שונים. הנח כי לפני בדיקת כל ביטוי מכיל המשתנה a את הערך 1, המשתנה b – את הערך 2, והמשתנה c – את הערך 3. מצא מהם ערכי המשתנים a, b ו-c לאחר בדיקת כל ביטוי, ומהו הערך המוחזר ממנו.

- | | |
|----------------------|--------------------------------|
| 1. a++ | 5. --b * -(a / c++) - 1 |
| 2. 3 + --c | 6. 7 * -c + ++b / a++ |
| 3. b++ + ++a | 7. (a + 1) * --c / (++b + a) |
| 4. c++ - a-- + ++b | 8. c--- --b |

22. מצא מהו הפלט המתקבל מהתוכנית הבאה, עבור כל אחת מקבוצות הקלטים המופיעות אחריה.

```
#include <stdio.h>
void main()
{
    int a, b;
    float c;

    printf("Please enter three numbers: ");
    scanf("%d %d %f", &a, &b, &c);

    printf("%d \n%f \n", a+++b, --c);
    printf("%f \n", ++b + 4.0 / (c - 1.0) + ++a);
    printf("%d \n", b++++-a);
    printf("%f \n", 4 % b+b / (b + 1-++c));
    printf("a = %d, b = %d, c = %f \n", a, b, c);
}
```

הערך המוחזר מאופרטורי השמה

23. לפני בדיקת כל אחד מהביטויים הבאים, ערכי המשתנים השלמים i ו-n הם 3 ו-5, בהתאמה. ערך המשתנה העשורוני f הוא 7.0. מצא מהם ערכיהם לאחר בדיקת כל ביטוי, ומהו הערך המוחזר ממנו.

- | | |
|--------------------|-------------------------|
| 1. i += 5 | 5. i /= (f = 3) + ++n |
| 2. f /= n | 6. i += f += n++ - 7; |
| 3. i = f = 4.9 | 7. i -= n =- f |
| 4. f *= n -= 4.0 | 8. f -= i = --n / 5 |

סוגי משתנים נוספים

24. לפניך מספר הוראות פלט וקלט. ציין עבור כל הוראה את סוג המשתנה אותו היא פולטת או קולטת.

1. `printf("%ld", a);`
2. `scanf("%lf", &b);`
3. `printf("%hd", c);`
4. `scanf("%Lf", &d);`
5. `printf("%Lf", e);`
6. `scanf("%hd", &f);`
7. `printf("%f", g);`
8. `scanf("%ld", &h);`

תשובות

.1

.4 ✓	.1 ✓
.5 x	.2 x
.6 x	.3 x

.2

1. 6	7. 1.0
2. 19.0	8. -7
3. 2	9. 39
4. 16	10. 3
5. -4	11. 49
6. 6	12. 0

.3

.5 אין להשתמש בחץ לביצוע ההשמה	.1 חסר נקודה-פסיק בסוף משפט ההשמה
.6 משמאל לסימן השוויון חייב להופיע משתנה אחד ויחיד	.2 אופרטור החזקה אינו קיים בשפת C (למעשה, משפט זה הוא חוקי, משום שהאופרטור \wedge הוא אופרטור XOR)
.7 אין להשתמש בסימן =: לביצוע ההשמה	.3 משמאל לסימן השוויון חייב להופיע משתנה אחד ויחיד
.8 אופרטור השורש אינו קיים בשפת C	.4 אין לבצע חלוקה באפס (זו אינה שגיאה תחבירית, אך חשוב לציין)

.4

.5 עשרוני	.1 שלם
.6 שלם	.2 עשרוני
.7 שלם	.3 עשרוני
.8 עשרוני	.4 שלם

.5

1.	35	8.	0
2.	8.0	9.	12300.0
3.	3	10.	0
4.	1	11.	3
5.	1.5	12.	2.0
6.	1.0	13.	2
7.	0.0	14.	990.0

.6

1. int i;
2. float age;
3. float salary, tax;
4. int exp, base;
5. int grade1, grade2, grade3;
6. float avg1, avg2, avg3;

.7

יש להפריד בין שמות המשתנים המוגדרים בפסיקים	.4	יש לציין את סוג המשתנה המוגדר	.1
אין להוסיף פסיק בין סוג המשתנה לשמות המשתנים המוגדרים	.5	על סוג המשתנה להופיע לפני שם המשתנה המוגדר	.2
אין להגדיר משתנים מסוגים שונים באותה שורת הגדרה	.6	יש לסיים את שורת ההגדרה בנקודה-פסיק	.3

.8

.2	i : 0	.1	i : 3
	x : -6		x : 6.0

1.	11 trees flourish at night
2.	Fighting ghosts on July 6th
3.	Neptune's orbital speed is 5.430000 km/s
4.	I mostly like channels 3 and 4
5.	6 pirates eating 2.500000 kids
6.	1 m/s is 2.237000 mi/h and 100.000000 cm/s

.10

1.	0.000000 trees flourish at night
2.	Fighting ghosts on July 0 th
3.	Neptune's orbital speed is km/s
4.	I mostly like channels 0 and 16512

.1 יש להחליף את תו ההמרה %f ל-d%

.2 יש לשנות את הפרמטר ל-6

.3 יש להוסיף את תו ההמרה %f לאחר המילה .is

.4 יש לשלוח להוראת הפלט שני פרמטרים נוספים: 3 ו-4

.11

אסור למחרוזת הבקרה של הוראת הקלט להכיל מלל המיועד לפליטה	.5 x	.1 ✓
שמות המשתנים שנשלחו להוראת הקלט לא הוקדמו בתו &	.6 x	.2 x לא נשלח משתנה עשרוני אל הוראת הקלט
	.7 ✓	.3 x אין התאמה בין תו ההמרה לבין הפרמטר המתאים לו
אין לרווח את תווי ההמרה	.8 x	.4 ✓

1.

```
#include <stdio.h>
void main()
{
    int x, y;

    printf("Please enter length and width of a rectangle: ");
    scanf("%d %d", &x, &y);

    printf("Area of the rectangle: %d \n", x * y);
    printf("Circumference of the rectangle: %d \n", 2 * (x + y));
}
```

2.

```
#include <stdio.h>
void main()
{
    float r;

    printf("Please enter a radius of a circle: ");
    scanf("%f", &r);

    printf("Area of the circle: %f \n", 3.14 * r * r);
    printf("Circumference of the circle: %f \n", 2.0 * 3.14 * r);
}
```

3.

```
#include <stdio.h>
void main()
{
    float speed, time;

    printf("Please enter speed [km/h] and time [h]: ");
    scanf("%f %f", &speed, &time);

    printf("Distance: %f [m] \n", 1000.0 * speed * time);
}
```

4.

```
#include <stdio.h>
void main()
{
    float mass;
```

```

printf("Please enter your mass [kg]: ");
scanf("%f", &mass);

printf("Mass in pounds: %f \n", mass * 2.205);
}

```

.13

92	.6 ✓	36	.1 ✓
אין לכתוב יותר מתו אחד בין	.7 ✗	אין לכתוב יותר מתו אחד בין	.2 ✗
הגרשיים (אלא אם מדובר בתו		הגרשיים (אלא אם מדובר בתו	
בקה כגון 'ח')		בקה כגון 'ח')	
הגרש שלאחר האלכסון ההפוך	.8 ✗	103	.3 ✓
נחשב, יחד עמו, לתו בקרה, ולא			
לגרש התוחם את התו			
100	.9 ✓	45	.4 ✓
אין לכתוב יותר מתו אחד בין	.10 ✗	אין לתחום את התו בגרשיים	.5 ✗
הגרשיים (אלא אם מדובר בתו		כפולים	
בקה כגון 'ח')			

.14

1.

```

#include <stdio.h>
void main()
{
    char ch;

    printf("Please enter a character: ");
    scanf("%c", &ch);

    printf("The following 4 chars are: %c %c %c %c \n",
          ch + 1, ch + 2, ch + 3, ch + 4);
}

```

2.

```

#include <stdio.h>
void main()
{
    char ch, delta;

    delta = 'a' - 'A';
    printf("Please enter a small English letter: ");
}

```

```

scanf("%c", &ch);

printf("The capital '%c' letter is '%c'. \n",
      ch, ch - delta);
}

```

3.

```

#include <stdio.h>
void main()
{
    /* letter1 --> capital letter
     * letter2 --> small letter
     */
    char letter1, letter2, digit1, digit2;

    printf("Please enter two English letters: ");
    scanf("%c %c", &letter1, &letter2);

    digit1 = letter1 - 'A' + '0';
    digit2 = letter2 - 'a' + '0';

    printf("Matching number: %c%c \n", digit1, digit2);
}

```

.15

– השורה החסרה

```
#include <conio.h>
```

.16

7.

```

#include <stdio.h>
#include <conio.h>
void main()
{
    char ch;

    printf("Please enter a character: ");
    ch = getch();

    printf("\nThe following 4 chars are: %c %c %c %c\n",
          ch + 1, ch + 2, ch + 3, ch + 4);
}

```

2.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char ch, delta;

    delta = 'a' - 'A';
    printf("Please enter a small English letter: ");
    ch = getch();

    printf("\nThe capital '%c' letter is '%c'. \n",
        ch, ch - delta);
}
```

3.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    /* letter1 --> capital letter
     * letter2 --> small letter
     */
    char letter1, letter2, digit1, digit2;

    printf("Please enter two English letters: ");
    letter1 = getch();
    letter2 = getch();

    digit1 = letter1 - 'A' + '0';
    digit2 = letter2 - 'a' + '0';

    printf("\nMatching number: %c%c \n", digit1, digit2);
}
```

.17

1.	1.23	5.	123.45
2.	4.000	6.	33333.0
3.	99.8	7.	111.222
4.	100	8.	76

.18

char ch1 = 'b'; int num1 = 10;	.5 X	float g = 9.78, g2 = 9.81;	.1 X
int room = 1, apart = 2;	.6 X		.2 ✓
	.7 ✓	int age = 25;	.3 X
int f1 = 0, f2 = 0;	.8 X		.4 ✓

.19

1. home += 10;	5. kg *= 3 + mass;
2. price *= 1.205;	6. min -= 1; או min--;
3. exp += 1; או exp++;	7. col %= 7 * row;
4. vol *= vol * vol;	8. month += 10 - tot;

.20

.6 ✓	.1 X אין להוסיף רווח בין תווי האופרטור המקוצר
.7 X אין לשלב שני אופרטורי העלאה עצמית באופן זה	.2 ✓
.8 ✓	.3 X אופרטור ההפחתה העצמית חייב להופיע לצד משתנה אחד ויחיד
.9 X אין לשלב שני אופרטורי הפחתה עצמית באופן זה	.4 X משמאל לאופרטור ההשמה המקוצר חייב להופיע משתנה אחד ויחיד
.10 X אופרטור ההעלאה העצמית חייב להופיע לצד משתנה אחד בלבד	.5 X תווי אופרטור ההשמה המקוצר מופיעים בסדר הפוך

.21

a: 1, b: 1, c: 4 ערך הביטוי: 1-	.5	a: 2, b: 2, c: 3 ערך הביטוי: 1	.1
a: 2, b: 3, c: 3 ערך הביטוי: 18-	.6	a: 1, b: 2, c: 2 ערך הביטוי: 5	.2
a: 1, b: 3, c: 2 ערך הביטוי: 1	.7	a: 2, b: 3, c: 3 ערך הביטוי: 4	.3
a: 1, b: 1, c: 2 ערך הביטוי: 2	.8	a: 0, b: 3, c: 4 ערך הביטוי: 5	.4

הפלט עבור $a=3, b=2, c=4.0$

הפלט עבור $a=5, b=5, c=10.0$

הפלט עבור $a=1, b=7, c=4.0$

5	עבור $a=3, b=2, c=4.0$
3.000000	
10.000000	
7	
4.000000	
$a = 4, b = 4, c = 4.000000$	
10	עבור $a=5, b=5, c=10.0$
9.000000	
13.500000	
12	
0.500000	
$a = 6, b = 7, c = 10.000000$	
8	עבור $a=1, b=7, c=4.0$
3.000000	
13.000000	
10	
5.500000	
$a = 2, b = 9, c = 4.000000$	

i: 0, n: 6, f: 3.0	.5	i: 8, n: 5, f: 7.0	.1
ערך הביטוי: 0		ערך הביטוי: 8	
i: 8, n: 6, f: 5.0	.6	i: 3, n: 5, f: 1.4	.2
ערך הביטוי: 8		ערך הביטוי: 1.4	
i: 10, n: -7, f: 7.0	.7	i: 4, n: 5, f: 4.9	.3
ערך הביטוי: 10		ערך הביטוי: 4	
i: 0, n: 4, f: 7.0	.8	i: 3, n: 1, f: 7.0	.4
ערך הביטוי: 7.0		ערך הביטוי: 7.0	

1.	long	5.	long double
2.	double	6.	short
3.	short	7.	float / double
4.	long double	8.	long

פרק 6

משפטי תנאי

שאלות

ביטויי יחס

1. סווג את ביטויי היחס הבאים לביטויים שגויים, לביטויים המחזירים ערך "אמת" ולביטויים המחזירים ערך "שקר":

1. $4 > 10$
2. $6 = 6 + 2$
3. $8 \neq 9 - 1$
4. $21 == 3 * 7$
5. $4 \% 8 > 8 / 4 + 2$

6. $6 <> 3 \% 2 * 2$
7. $9 / 3 >= 0 + 1 + 1 + 1$
8. $6 / (2 - 1) = 18 / 3$
9. $10 / 5 / 1 \neq 10 \% 4$
10. $-(7 + 1) \leq -5$

משפט התנאי if

2. אחדים ממשפטי ה-if שלפניך שגויים. סמן אותם והסבר מהן הטעויות.

```
1.  
if (avg > 100)  
    printf("Wrong values entered!");
```

```
2.  
if weight <= 44.5  
    printf("You are very thin");
```

```
3.  
IF (exp == 0)  
    printf("Result is definitely 1");
```

```
4.  
if (b != 0.0)  
    printf("a / b is: %f", a / b);
```

```
5.  
if (num % 2 == 0);  
    printf("The number is even.");
```

```
6.  
if (mice > cats)  
    printf("Beware, oh furry creatures!");
```

```
7.  
if (gender >= 1)  
    scanf("%d", &gender)
```

```
8.  
if (depth > 25)  
    /* very, very deep! */  
    printf("Deeeep ! ! !");
```

3. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש שני ערכים חיוביים, המייצגים את אורך צלעותיו של מלבן, ופולטת הודעה מתאימה אם המלבן הוא ריבוע.

2. תוכנית הקולטת מהמשתמש שני מספרים טבעיים ופולטת הודעה מתאימה אם הראשון הוא כפולה של השני.

3. תוכנית הקולטת מהמשתמש ארבעה מספרים ופולטת את הגדול מביניהם.

משפט התנאי if-else

4. ציין, עבור כל אחד ממשפטי if-else הבאים :

- ❖ מהו הפלט (אם יש) כאשר ביטוי התנאי מתקיים וכאשר הוא אינו מתקיים.
- ❖ כמה משפטים שייכים למשפט ה-if.

1.

```
if (sales <= 125000.0) {  
    printf("Business is ");  
    printf("bad");  
}
```

2.

```
printf("I'm ");  
if (year > 1983)  
    printf("not ");  
    printf("old enough");
```

3.

```
if (height > 1.90) {  
    printf("Very tall");  
    height = 1.0;  
}  
else  
    printf("Not so tall");
```

4.

```
if (class == 4)  
    printf("Great");  
else  
    printf("Or not");  
    printf("!!!");
```

5. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מספר טבעי ופולטת הודעה מתאימה אם הוא מתחלק ב-3 ללא שארית.
2. תוכנית הקולטת מהמשתמש שני מספרים ותו. ידוע כי ערך התו הוא '+' או '-'. אם הוא '+', ייפלט סכום שני המספרים; אם הוא '-', ייפלט הפרש שני המספרים.
3. תוכנית הקולטת מהמשתמש מספר שלם בעל ארבע ספרות ותו. ידוע כי ערך התו הוא '3' או '4'. אם הוא '3', תיפלט ספרת המאות של המספר; אם הוא '4', תיפלט ספרת האלפים של המספר.

קינון משפטי if

6. כתוב את המשפטים הבאים תוך הזחתם ימינה, כך שיהיה ברור לעין לאילו משפטים הם שייכים:

1.

```
if (a > b)
printf("a > b");
else
printf("a <= b");
```

2.

```
if (grade1 > grade2) {
max = grade1;
printf("%d", grade1);
} if (grade1 == grade2)
printf("Equals!");
if (grade1 < grade2) {
max = grade2;
printf("%d", grade2);
}
```

3.

```
if (x == i)
if (i == j)
printf("x == i, i == j");
```

4.

```
if (l > m)
if (m > n)
printf("l > n");
printf("\n");
```

5.

```
if (num > 100) {
printf("100 < num ");
if (num < 1000)
printf("< 1000");
}
```

6.

```
if (n % 2 == 0) {
printf("n is even \n");
if (n % 3 == 0)
if (n % 5 == 0)
printf("n % 30 == 0");
}
```

7. כתוב את התוכניות הבאות :

1. תוכנית הקולטת אות קטנה מהמשתמש ופולטת הודעה מתאימה אם היא אינה אם-קריאה (אמות-הקריאה באנגלית הן האותיות 'a', 'e', 'i', 'o' ו-'u').
2. תוכנית הקולטת מהמשתמש ארבעה מספרים שלמים ופולטת הודעה מתאימה אם הם שווים זה לזה.
3. תוכנית הקולטת מספר טבעי מהמשתמש ופולטת הודעות מתאימות עבור המקרים הבאים:
 - ❖ המספר בין 10 ל-20 (כולל)
 - ❖ המספר בין 10 ל-20 (כולל) והמספר מתחלק ב-3 ללא שארית
 - ❖ המספר בין 10 ל-15 (כולל) והמספר מתחלק ב-3 ללא שאריתהשתמש בכמה שפחות הוראות if (שים לב כי התוכנית יכולה לפלוט יותר מהודעה אחת עבור מספר נתון).

קינון משפטי if-else

8. לפניך מספר קטעי תוכניות. ציין מהו הפלט (אם יש) המתקבל מכל אחד מהם, עבור כל אחד מהערכים הרשומים אחריו (שים לב כי מספר טעויות נעשו בהזחת המשפטים).

1.

```
if (x > 4)
    if (x > 7)
        printf("*");
else
    printf("-");
```

הפלט עבור $x=10$:

הפלט עבור $x=0$:

הפלט עבור $x=5$:

2.

```
if (x % 3 == 0)
    if (x / 4 != 6)
        if (x < 20)
            printf("-");
else
    printf("x");
```

הפלט עבור $x=0$:

הפלט עבור $x=24$:

הפלט עבור $x=36$:

3.

```
if (x < 0) {  
    if (x > -8)  
        printf("*");  
}  
else  
    printf("-");
```

הפלט עבור $x=-7$:

הפלט עבור $x=3$:

הפלט עבור $x=-9$:

4.

```
if (x > 100)  
    printf("*");  
else if (x > 10)  
    printf("+");  
else if (x > 0)  
    printf("/");  
else if (x < 0)  
    printf("-");  
else  
    printf(".");
```

הפלט עבור $x=140$:

הפלט עבור $x=3$:

הפלט עבור $x=0$:

הפלט עבור $x=-4$:

הפלט עבור $x=13$:

הפלט עבור $x=100$:

5.

```
if (x % 2 == 0)  
    if (x % 3 == 0)  
        if (x % 5 == 0)  
            if (x % 7 == 0)  
                printf("+");
```

```
        else
            printf("-");
else
    printf("*");
```

הפלט עבור x=0

הפלט עבור x=90

הפלט עבור x=210

הפלט עבור x=1

הפלט עבור x=12

הפלט עבור x=4

המשפט switch

9. אחדים ממשפטי ה-switch שלפניך שגויים. סמן אותם והסבר מהן הטעויות.

1.

```
switch (value) {
case num1:
    printf("value equals num1);
    break;
case num2:
    printf("value equals num2");
    break;
}
```

2.

```
switch (13) {
case 11:
    printf("Eleven");
    break;
case 12:
    printf("Twelve");
    break;
case 13:
    printf("Thirteen");
    break;
}
```

3.

```
switch (g) {
case 9.78:
    printf("free-fall acceleration on Earth");
    break;
case 0.5:
    printf("free-fall acceleration on Pluto");
    break;
case 3.72:
    printf("free-fall acceleration on Mars");
    break;
}
```

4.

```
switch (age) {
case 13:
    printf("You're thirteen years old. ");
    break;
case 16:
    printf("You're sixteen years old. ");
    break;
case 13:
    printf("Have you had your bar-mitsva yet? ");
    break;
}
```

5.

```
switch (n % 2) {
case 0:
    printf("%d is even \n", n);
    break;
case 1:
    printf("%d is odd \n", n);
    break;
}
```

6.

```
switch x {
case 'd':
    printf("integer");
    break;
case 'f':
    printf("floating point");
    break;
case '%':
    printf("percentage");
    break;
}
```

7.

```
switch (month) {
case 4:
    printf("April");
    break;
}
```

8.

```
switch (day)
case 1:
    printf("Sunday");
    break;
case 2:
    printf("Monday");
    break;
case 3:
    printf("Tuesday");
    break;
case 4:
    printf("Wednesday");
    break;
```

10. לפניך מספר משפטי if ו-switch. במידת האפשר, הפוך את משפטי ה-if למשפטי switch. אם לא ניתן לבצע זאת, הסבר מדוע.

1.

```
if (a == 2)
    printf("a == 2");
else if (a == 3)
    printf("a == 3");
else if (a == 4)
    printf("a == 4");
```

2.

```
if (num > 0)
    printf("num is positive");
else if (num < 0)
    printf("num is negative");
```

3.

```
if (a == b)
    printf("a equals b");
else if (a == c)
    printf("a equals c");
```

4.

```
if (num == 0) {
    max = 25;
    min = 10;
    printf("num equals zero");
}
else if (another_num == 1) {
    max = 10;
    min = 7;
    printf("another_num equals one");
}
```

5.

```
if (fact == 720)
    n = 6;
else if (fact == 120)
    n = 5;
else if (fact == 24)
    n = 4;
else if (fact == 6)
    n = 3;
```

6.

```
if (q == 7.5)
    printf("7.5");
else if (q == 6.2)
    printf("6.2");
```

11. אחדות משורות ה-case שלפניך שגויות. סמן אותן והסבר מהן הטעויות.

- | | | | |
|----|---------------------|----|---------------|
| 1. | case x: | 5. | case 12.5: |
| 2. | case 7 * 8 / 4 / 2: | 6. | case 'q' + 3: |
| 3. | case 4, 5, 6: | 7. | Case 999: |
| 4. | case 100: | 8. | case 0.0: |

12. לפניך מספר משפטי if ו-switch. הפוך את משפטי ה-if למשפטי switch מתאימים (תוך שימוש בגלישה) ואת משפטי ה-switch למשפטי if מתאימים.

1.

```
if (a == 12 || a == 16)
    printf("a / 4: %d", a / 4);
else if (a == 9)
    printf("a / 3: 3");
```

2.

```
switch (c) {
case 'a':
case 'e':
case 'i':
case 'o':
case 'u':
    printf("vowel!");
    break;
case '1':
    printf("digit!");
}
```

3.

```
if (num == 131 || num == 9 || num == 14741)
    printf("palindrome!");
else if (num == 3 || num == 5 || num == 7)
    printf("prime!");
else if (num == 0)
    printf("zero!");
```

4.

```
switch (quad) {
case 1:
case 2:
case 3:
    printf("quad <= 3 \n");
    if (quad == b)
        printf("quad == b");
    break;
}
```

5.

```
if (ch == '!') {
    printf("Hello ");
    ch = '*';
}
if (ch == '*') {
    printf("John. ");
    ch = '?';
}
if (ch == '?')
    printf("How are you? ");
```

שורת ה-default

13. מצא מהו הפלט המתקבל מכל אחד ממשפטי switch הבאים, עבור כל אחד מהקלטים המופיעים אחריו.

1.

```
switch (x) {
case 1:
    printf("1");
    break;
case 2:
    printf("2");
default:
    printf("-");
}
```

הפלט עבור x=1:

הפלט עבור x=2:

הפלט עבור x=4:

2.

```
switch (x / 5) {
case 3:
    printf("*");
default:
    printf("+");
case 1:
    printf("_");
}
```

הפלט עבור x=5:

הפלט עבור x=4:

הפלט עבור x=18:

3.

```
switch (x % 4) {
default:
    printf("#");
    x = 3;
    break;
case 2:
    printf("$");
case 3:
    printf("_");
}
```

:x=7 הפלט עבור

:x=10 הפלט עבור

:x=4 הפלט עבור

4.

```
switch (x + 2 * x + 2) {
case 5:
    printf("[");
case 11:
    printf("^");
default:
    printf("]");
    break;
    printf("!");
}
```

:x=1 הפלט עבור

:x=3 הפלט עבור

:x=2 הפלט עבור

5.

```
switch (x + y) {
default:
    if (x > y)
        printf(">");
    else
        printf("<=");
    printf("&");
case 4 + 6:
    printf("%d", x + y * 2);
    break;
}
```

```
case 2 - 9:
    printf("%d", y);
    break;
}
```

הפלט עבור $x=1, y=2$:

הפלט עבור $x=3, y=7$:

הפלט עבור $x=-11, y=4$:

14. כתוב את התוכניות הבאות:

- בתפריט ראשי של תוכנה נתונות האפשרויות הבאות: (0) – יציאה; (1) – פתיחת קובץ; (2) – שמירת קובץ; (3) – הדפסת קובץ (המספרים שבסוגריים מציינים את מספר האפשרות). כתוב תוכנית הפולטת את תפריט התוכנה, קולטת מהמשתמש תו, המייצג את מספר האפשרות המבוקשת, ופולטת את הפעולה שעתידה להתבצע. אם יוזן תו שאינו מתאים לאחד האפשרויות, תוצג הודעה מתאימה.
- תוכנית הקולטת מהמשתמש שני מספרים חיוביים ותו. התו יכול להיות אחד מהתווים הבאים: '+', '-', '*' ו-'/'. התוכנית תבצע את פעולת התו בין שני המספרים ותפלוט את התוצאה. אם יוזן תו שאינו אחד מהתווים הללו, תבוצע פעולת כפל בין השניים.
- תוכנית הקולטת מהמשתמש מספר טבעי ובודקת מהי ספרתו הימנית ביותר. אם ספרה זו זוגית, היא תיפלט מספר פעמים השווה לה. אם היא אי-זוגית, ייפלט המספר מספר פעמים השווה לה.

אופרטורים לוגיים

אופרטור וגם הלוגי

15. כתוב את התוכניות הבאות:

- תוכנית הקולטת מהמשתמש שלושה מספרים שונים ופולטת הודעה מתאימה אם הם הוזנו בסדר עולה.
- תוכנית הקולטת ארבע ספרות מהמשתמש (כארבעה תווים נפרדים) המייצגות מספר בעל ארבע ספרות. התוכנית תפלוט את המלל "Match!" אם הספרות מהוות סדרה חשבונית; כלומר, אם ההפרש בין כל שתי ספרות סמוכות זהה.
- תוכנית הקולטת שלוש אותיות מהמשתמש ופולטת הודעה מתאימה אם כולן אותיות קטנות, כולן אותיות גדולות או חלקן קטנות וחלקן גדולות.

אופרטור או הלוגי

16. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספר טבעי קטן מ-10 ומציגה הודעה מתאימה אם הוא מספר ראשוני.
2. תוכנית הקולטת תו מהמשתמש, המייצג אופרטור בודד בשפת C, ותפלוט את קבוצת האופרטורים אליה הוא שייך. כזכור, האופרטורים < ו- > הם אופרטורי יחס והאופרטורים +, -, *, / ו-% הם אופרטורים אריתמטיים (שים לב שאופרטורים המורכבים משני תווים, כגון <= ו- &&, אינם מוזכרים).
3. תוכנית הקולטת מהמשתמש אורך צלעותיו של משולש ומציגה הודעה מתאימה אם הוא משולש ישר-זווית או לא, תוך שימוש במשפט פיתגורס. שים לב, שלא ידוע באיזה סדר יזין המשתמש את אורך הצלעות.

אופרטור השלילה הלוגי

17. לפניך מספר ביטויים. כתוב ביטוי שקול לכל אחד מהם תוך שימוש באופרטור השלילה.

- | | |
|---------------------------------|--|
| 1. $x == 3$ | 4. $x > 10 \ \&\& \ x < 15$ |
| 2. $n \geq m$ | 5. $x \neq 7 \ \&\& \ x \geq 0 \ \&\& \ x < 25$ |
| 3. $ch == 'a' \ \ ch == 'b'$ | 6. $y == 2.25 \ \ y == 6.25$
$\ \ y == 12.25$ |

שילוב אופרטורים לוגיים

18. אחדים מהביטויים הלוגיים שלפניך שגויים. סמן אותם והסבר מהן הטעויות.

1. $x > 10 \ \&\& \ x < 100$
2. $num == 0.0 \ || \ num > 0.0 \ || \ num < 0.0$
3. $!(width \geq q) \ || \ width = < 0$
4. $field \neq 10.0 \ || \ field = = 9.0$
5. $(day == (17)) \ \&\& \ month == 4 \ \&\& \ year == 1983$
6. $(weight > 48)!$
7. $(divider1 * divider2 * divider3) \neq 0$
8. $(b * b) - 4.0 * (a * c)$
9. $!(grade < 0 \ || \ grade > 100 \ || \ grade == avg)$
10. $(height < 0.0) \ ! \ (height > 1.80)$

19. קצר כל אחד מן הביטויים הלוגיים הבאים, כך שיכילו כמה שפחות אופרטורים. התייחס למשתנה x כאל משתנה שלם.

1. $x < 0 \ || \ x > 0$
2. $x == 4 \ || \ x == 5 \ || \ x == 6 \ || \ x == 7$
3. $x > 10 \ \&\& \ x != 11$
4. $!(x == 0 \ || \ x > 0)$
5. $(x > 1 \ \&\& \ x < 10) \ || \ (x >= 10 \ \&\& \ x < 20)$
6. $x == 0 \ || \ x == 2 \ || \ x == 4 \ || \ x == 6 \ || \ x == 8 \ || \ x == 10$

20. כתוב מחדש את התוכניות המופיעות בתרגיל 7, תוך צמצום מספר הוראות התנאי בכל תוכנית על ידי שימוש באופרטורים לוגיים מתאימים.

ערכי ביטויי יחס

21. מצא את ערך הביטויים הבאים:

1. $3 != 2 + 1$
2. $(3 != 2) + 1$
3. $!1 - 1$
4. $!(1 - 1)$
5. $'a' == 'b' - 1 \ \&\& \ 'd' == 'c' - 1$
6. $4 >= 6 \ || \ num = 15 / 10$
7. $10 == 10 \% 20 \ || \ 7 > 6 \ \&\& \ 3 < -5$
8. $var = 10 <= 20 / 2 \% 5 \ || \ 0 / 3.4$

22. מצא מהו הפלט המתקבל מהתוכנית הבאה, עבור כל אחת מקבוצות הקלטים המופיעות אחריה.

```
#include <stdio.h>
void main()
{
    int a, b, c;

    printf("Please enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    printf("a > b && ++c == b: %d \n", a > b && ++c == b);
    printf("!!!(a-- == --b): %d \n", !!!(a-- == --b));
    printf("!(b *= b / (b + b)): %d \n", !(b *= b / (b + b)));
    printf("!(--a > 0) && (b += -c) < c: %d \n",
           !(--a > 0) && (b += -c) < c);
}
```

עבור a=4, b=6, c=3

עבור a=2, b=3, c=0

עבור a=2, b=-4, c=5

23. מצא את ערך הביטויים הבאים:

1. $(\text{num} = 10) \ || \ !x$
2. $4 == 4 == 1$
3. $9 > 6 > 3$
4. $0 \ \&\& \ 3 \ || \ 6$

5. $-7 < -5 < -3$
6. $!5 \ || \ !0 \ \&\& \ !5$
7. $3 == 3 == 3$
8. $!(1 - !(1 - !(0)))$

תשובות

.1

.6 x	שקר	.1 ✓
.7 ✓	אמת	.2 x
.8 x	שקר	.3 ✓
.9 ✓	אמת	.4 ✓
.10 x	שקר	.5 ✓

.2

.5 x	אין להוסיף נקודה-פסיק לאחר הסוגריים הסוגרים של ביטוי התנאי (זו אינה שגיאה תחבירית, אך יש להתייחס אליה ככזו)	.1 ✓
.6 ✓	יש לתחום את ביטוי התנאי בסוגריים	.2 x
.7 x	יש להוסיף נקודה-פסיק לאחר המשפט השייך למשפט ה-if	.3 x
.8 ✓	את המילה השמורה if יש לכתוב באותיות לועזיות קטנות	.4 ✓

.3

1.

```
#include <stdio.h>
void main()
{
    float length, width;

    printf("Please enter length and width: ");
    scanf("%f %f", &length, &width);

    if (length == width)
        printf("It's a square! \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    int num1, num2;

    printf("Please enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    if (num1 % num2 == 0)
        printf("%d %% %d is 0! \n", num1, num2);
}
```

3.

```
#include <stdio.h>
void main()
{
    float num1, num2, num3, num4, highest;

    printf("Please enter four numbers: ");
    scanf("%f %f %f %f", &num1, &num2, &num3, &num4);

    highest = num1;
    if (num2 > highest)
        highest = num2;
    if (num3 > highest)
        highest = num3;
    if (num4 > highest)
        highest = num4;

    printf("The highest number is: %f \n", highest);
}
```

.4

3. כאשר ביטוי התנאי מתקיים: Very tall כאשר ביטוי התנאי אינו מתקיים: Not so tall	1. כאשר ביטוי התנאי מתקיים: Business is bad כאשר ביטוי התנאי אינו מתקיים: (אין פלט)
4. כאשר ביטוי התנאי מתקיים: Great!!! כאשר ביטוי התנאי אינו מתקיים: Or not!!!	2. כאשר ביטוי התנאי מתקיים: I'm not old enough כאשר ביטוי התנאי אינו מתקיים: I'm old enough

1.

```
#include <stdio.h>
void main()
{
    int num;

    printf("Please enter a number: ");
    scanf("%d", &num);

    if (num % 3 == 0)
        printf("num % 3 == 0 \n");
    else
        printf("num % 3 != 0 \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    float num1, num2, output;
    char ch;

    printf("Please enter two numbers and a char: ");
    scanf("%f %f %c", &num1, &num2, &ch);

    if (ch == '+')
        output = num1 + num2;
    else
        output = num1 - num2;

    printf("%f %c %f = %f \n", num1, ch, num2, output);
}
```

3.

```
#include <stdio.h>
void main()
{
    int num, digit;
    char ch;

    printf("Please enter a number and a char: ");
    scanf("%d %c", &num, &ch);
}
```

```

if (ch == '3') {
    digit = num / 100;
    digit %= 10;
}
else
    digit = num / 1000;

printf("The digit is: %d \n", digit);
}

```

.6

1.

```

if (a > b)
    printf("a > b");
else
    printf("a <= b");

```

2.

```

if (grade1 > grade2) {
    max = grade1;
    printf("%d", grade1);
}
if (grade1 == grade2)
    printf("Equals!");
if (grade1 < grade2) {
    max = grade2;
    printf("%d", grade2);
}

```

3.

```

if (x == i)
    if (i == j)
        printf("x == i, i == j");

```

4.

```

if (l > m)
    if (m > n)
        printf("l > n");
printf("\n");

```

5.

```

if (num > 100) {
    printf("100 < num ");
}

```

```
    if (num < 1000)
        printf("< 1000");
}
```

6.

```
if (n % 2 == 0) {
    printf("n is even \n");
    if (n % 3 == 0)
        if (n % 5 == 0)
            printf("n % 30 == 0");
}
```

.7

1.

```
#include <stdio.h>
void main()
{
    char letter;

    printf("Please enter a small English letter: ");
    scanf("%c", &letter);

    if (letter != 'a')
        if (letter != 'e')
            if (letter != 'i')
                if (letter != 'o')
                    if (letter != 'u')
                        printf("The letter is not a vowel. \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    int num1, num2, num3, num4;

    printf("Please enter four numbers: ");
    scanf("%d %d %d %d", &num1, &num2, &num3, &num4);

    if (num1 == num2)
        if (num2 == num3)
            if (num3 == num4)
                printf("The numbers are equal! \n");
}
```

3.

```
#include <stdio.h>
void main()
{
    int num;

    printf("Please enter a number: ");
    scanf("%d", &num);

    if (num >= 10)
        if (num <= 20) {
            printf("num is between 10 and 20. \n");
            if (num % 3 == 0) {
                printf("num % 3 == 0. \n");
                if (num <= 15)
                    printf("num is between 10 and 15. \n");
            }
        }
}
```

.8

1.

*	עבור x=10
(אין פלט)	עבור x=0
-	עבור x=5

2.

-	עבור x=0
(אין פלט)	עבור x=24
x	עבור x=36

3.

*	עבור x=-7
-	עבור x=3
(אין פלט)	עבור x=-9

4.

*	עבור $x=140$
/	עבור $x=3$
.	עבור $x=0$
-	עבור $x=-4$
+	עבור $x=13$
+	עבור $x=100$

5.

+	עבור $x=0$
-	עבור $x=90$
+	עבור $x=210$
(אין פלט)	עבור $x=1$
*	עבור $x=12$
(אין פלט)	עבור $x=4$

.9

.5 ✓	ערכי שורות ה-case חייבים להיות קבועים	.1 ✗
.6 ✗	את הביטוי הנבדק בראש משפט ה-switch יש לתחום בסוגריים	.2 ✓
.7 ✓	ערכי שורות ה-case חייבים להיות שלמים	.3 ✗
.8 ✗	את שורות ה-case ואת המשפטים שלהן יש לתחום בסוגריים מסולסלים	.4 ✗

1.

```
switch (a) {
case 2:
    printf("a == 2");
    break;
case 3:
    printf("a == 3");
    break;
case 4:
    printf("a == 4");
    break;
}
```

2.

לא ניתן להפוך למשפט switch, משום שבמשפט switch ניתן לבדוק רק יחס שוויון.

3.

לא ניתן להפוך למשפט switch, משום שבמשפט switch לא ניתן להשוות את ערך הביטוי הנבדק עם ערך משתנה.

4.

לא ניתן להפוך למשפט switch, משום שבמשפט switch ניתן לבדוק רק ערך משתנה בודד (ולא של שניים, כפי שנעשה בקטע התוכנית).

5.

```
switch (fact) {
case 720:
    n = 6;
    break;
case 120:
    n = 5;
    break;
case 24:
    n = 4;
    break;
case 6:
    n = 3;
    break;
}
```

6.

לא ניתן להפוך למשפט switch, משום שבמשפט switch לא ניתן לבדוק ערכים עשרוניים.

.11

ערך שורת ה-case חייב להיות שלם	.5 x	ערך שורת ה-case חייב להיות קבוע	.1 x
	.6 ✓		.2 ✓
יש לכתוב את המילה השמורה case באותיות קטנות	.7 x	לצד המילה השמורה case חייב להופיע קבוע אחד בלבד	.3 x
ערך שורת ה-case חייב להיות שלם	.8 x	יש להוסיף נקודתיים לאחר שורת ה-case	.4 x

.12

1.

```
switch (a) {
case 12:
case 16:
    printf("a / 4: %d", a / 4);
    break;
case 9:
    printf("a / 3: 3");
    break;
}
```

2.

```
if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
    printf("vowel!");
else if (c == '1')
    printf("digit!");
```

3.

```
switch (num) {
case 131:
case 9:
case 14741:
    printf("palindrome!");
    break;
case 3:
case 5:
case 7:
    printf("prime!");
    break;
case 0:
    printf("zero!");
    break;
}
```

4.

```
if (quad == 1 || quad == 2 || quad == 3) {  
    printf("quad <= 3 \n");  
    if (quad == b)  
        printf("quad == b");  
}
```

5.

```
switch (ch) {  
    case '!':  
        printf("Hello ");  
    case '*':  
        printf("John. ");  
    case '?':  
        printf("How are you? ");  
        break;  
}
```

.13

1.

1	עבור x=1
2-	עבור x=2
-	עבור x=4

2.

-	עבור x=5
+_	עבור x=4
*+_	עבור x=18

3.

-	עבור x=7
\$_	עבור x=10
#	עבור x=4

4.

[^]	עבור x=1
^]	עבור x=3
]	עבור x=2

5.

<=&5	עבור x=1, y=2
17	עבור x=3, y=7
4	עבור x=-11, y=4

.14

1.

```
#include <stdio.h>
void main()
{
    char option;

    printf("Select one of the following: \n");
    printf("\t(0) - Quit \n\t(1) - Open file \n");
    printf("\t(2) - Save file \n\t(3) - Print file \n");
    scanf("%c", &option);

    switch (option) {
    case '0':
        printf("Quiting program... \n");
        break;
    case '1':
        printf("Opening file... \n");
        break;
    case '2':
        printf("Saving file... \n");
        break;
    case '3':
        printf("Printing file... \n");
        break;
    default:
        printf("Invalid choice! \n");
    }
}
```

2.

```
#include <stdio.h>
void main()
{
    float num1, num2, result;
    char op;

    printf("Please enter two numbers and a char: ");
    scanf("%f %f %c", &num1, &num2, &op);
```

```

switch (op) {
case '-':
    result = num1 - num2;
    break;
case '+':
    result = num1 + num2;
    break;
default:
    printf("Invalid choice! multiplying as default. \n");
    op = '*';
case '*':
    result = num1 * num2;
    break;
case '/':
    result = num1 / num2;
    break;
}

printf("%f %c %f = %f \n", num1, op, num2, result);
}

```

3.

```

#include <stdio.h>
void main()
{
    int num, digit;

    printf("Please enter a number: ");
    scanf("%d", &num);

    digit = num % 10;
    switch (digit) {
case 0:
case 2:
case 4:
case 6:
case 8:
        num = digit;
default:
        for ( ; digit > 0 ; digit--)
            printf("%d ", num);
    }
}

```

1.

```
#include <stdio.h>
void main()
{
    float num1, num2, num3;

    printf("Please enter three numbers: ");
    scanf("%f %f %f", &num1, &num2, &num3);

    if (num1 < num2 && num2 < num3)
        printf("Values were entered in ascending order. \n");
    else
        printf("Values weren't entered in ascending order. \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    char dig1, dig2, dig3, dig4;

    printf("Please enter four digits: ");
    scanf("%c %c %c %c", &dig1, &dig2, &dig3, &dig4);

    if (dig3 - dig2 == dig2 - dig1 && dig4 - dig3 == dig3 - dig2)
        printf("Match! \n");
}
```

3.

```
#include <stdio.h>
void main()
{
    char let1, let2, let3;

    printf("Please enter three letters: ");
    scanf("%c %c %c", &let1, &let2, &let3);

    if (let1 >= 'a' && let1 <= 'z' &&
        let2 >= 'a' && let2 <= 'z' &&
        let3 >= 'a' && let3 <= 'z')
        printf("All the letters are small letters! \n");
    else if (let1 >= 'A' && let1 <= 'Z' &&
             let2 >= 'A' && let2 <= 'Z' &&
             let3 >= 'A' && let3 <= 'Z')
        printf("All the letters are capital letters! \n");
}
```

```
    else
        printf("The letters are small and capital letters! \n");
}
```

.16

1.

```
#include <stdio.h>
void main()
{
    int num;

    printf("Please enter a number: ");
    scanf("%d", &num);

    if (num == 2 || num == 3 || num == 5 || num == 7)
        printf("The number is prime! \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    char op;

    printf("Please enter a C operator: ");
    scanf("%c", &op);

    if (op == '>' || op == '<')
        printf("You've entered a relational operator. \n");
    else if (op == '+' || op == '-' || op == '*' || op == '/' ||
             op == '%')
        printf("You've entered an arithmetic operator. \n");
}
```

3.

```
#include <stdio.h>
void main()
{
    float a, b, c;

    printf("Please enter lengths of triangle sides: ");
    scanf("%f %f %f", &a, &b, &c);
}
```

```

if (a * a == b * b + c * c ||
    b * b == a * a + c * c ||
    c * c == a * a + b * b)
    printf("One angle is 90 degrees. \n");
else
    printf("No angle is 90 degrees. \n");
}

```

.17

1. $!(x != 3)$	4. $!(x <= 10 \ \ x >= 15)$
2. $!(n < m)$	5. $!(x == 7 \ \ x < 0 \ \ x >= 25)$
3. $!(ch != 'a' \ \&\& \ ch != 'b')$	6. $!(y != 2.25 \ \&\& \ y != 6.25 \ \&\& \ y != 12.25)$

.18

<p>6 x על אופרטור השלילה הלוגי להופיע משמאל לביטוי ולא מימינו</p>	<p>1 x אין להוסיף רווח בין שני תווי ה- & של אופרטור הוגם הלוגי</p>
<p>7 x על התו ! שמרכיב את אופרטור היחס "שונה מ..." להופיע לפני התו = (למעשה, ביטוי זה מנסה להציב את הערך הלוגי ההפוך מ-0 (!0) במכפלת שלושת המשתנים, והדבר אסור)</p>	<p>2 ✓</p>
<p>8 x הופיעו סוגריים פותחים עבור הביטוי $a * c$, אך לא הופיעו סוגריים סוגרים</p>	<p>3 x על התו < שמרכיב את אופרטור היחס "קטן או שווה ל..." להופיע לפני התו =</p>
<p>9 ✓</p>	<p>4 x אין להוסיף רווח בין שני התווים המרכיבים את אופרטור השוויון ואת אופרטור האי-שוויון</p>
<p>10 x על אופרטור השלילה הלוגי להופיע משמאל לביטוי ולא בין שני ביטויים</p>	<p>5 ✓</p>

1.	<code>x != 0</code>	4.	<code>x < 0</code>
2.	<code>x >= 4 && x <= 7</code>	5.	<code>x > 1 && x < 20</code>
3.	<code>x > 11</code>	6.	<code>x >= 0 && x <= 10 && x % 2 == 0</code>

1.

```
#include <stdio.h>
void main()
{
    char letter;

    printf("Please enter a small English letter: ");
    scanf("%c", &letter);

    if (letter != 'a' && letter != 'e' && letter != 'i' &&
        letter != 'o' && letter != 'u')
        printf("The letter is not a vowel. \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    int num1, num2, num3, num4;

    printf("Please enter four numbers: ");
    scanf("%d %d %d %d", &num1, &num2, &num3, &num4);

    if (num1 == num2 && num2 == num3 && num3 == num4)
        printf("The numbers are equal! \n");
}
```

3.

```
#include <stdio.h>
void main()
{
    int num;

    printf("Please enter a number: ");
    scanf("%d", &num);
}
```


פרק 7

לולאות

שאלות

הלולאה while

1. אחדים מקטעי התוכניות שלפניך שגויים. סמן אותם והסבר מהן הטעויות. עבור קטעי התוכניות הנכונים, ציין מהו הפלט המתקבל מהם.

```
1.  
int num = 0;  
while (num < 3) {  
    printf("* ");  
    num++;  
}
```

```
2.  
int counter = 1;  
while counter != 5 {  
    printf("Loop! \n");  
    counter++;  
}
```

```
3.  
int from = 5, to = 2;  
while (from > to) {  
    printf("hello, world \n");  
    from--;  
}
```

4.

```
int zug1 = 2;
WHILE (zug1 < 20) {
    printf("%d ", zug1);
    zug1 += zug1;
}
```

5.

```
int head = 0, tail = 60;
while (head != tail) {
    printf("%d %d ", head, tail);
    head += 10;
    tail -= 5;
}
```

6.

```
int dist = 3, fair = 15;
while (dist <= fair) {
    dist %= 2;
    dist += 2;
    fair -= dist;
    printf("-%-d- ", fair % 2);
}
```

7.

```
int true = 1;
while (true > 0 && true < 10) {
    true %= true + 1;
    true += true;
    printf("%d ", 3 * true / 2);
}
```

8.

```
int worm = 0, warm = 5;
while (worm < warm || worm == 3) {
    warm -= warm % warm + 1;
    worm = worm - warm + 1 + warm * (warm / warm);
    printf("%d%d", warm, worm);
}
```

2. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש שני מספרים טבעיים ומחשבת את מכפלתם, מבלי להשתמש בפעולת הכפל (*).
2. תוכנית הקולטת מהמשתמש שני מספרים טבעיים ומחשבת את מנת הראשון בשני, מבלי להשתמש בפעולת החילוק (/). ידוע כי המספר הראשון גדול מהמספר השני והוא מתחלק בו ללא שארית.
3. תוכנית הקולטת מהמשתמש שני מספרים טבעיים ומחשבת את שארית המספר הראשון בשני, מבלי להשתמש בפעולת השארית (%).

3. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש שתי אותיות לועזיות (שתייהן קטנות או שתייהן גדולות), כאשר הראשונה קודמת לשנייה בסדר האלפביתי, ופולטת את כל האותיות שבין שתייהן (כולל אותן).
2. e הוא קבוע מתמטי שערכו, בדיוק של שש ספרות אחרי הנקודה העשרונית, הוא 2.718281. ערך הביטוי $(1 + \frac{1}{n})^n$ שואף ל- e ככל ש- n (מספר טבעי) גדול יותר. חשב את ערך e בדיוק סביר (בתחילה, כאשר n הוא 100, ואחר כך, כאשר n הוא 1000 ו-10000).
3. תוכנית הקולטת מהמשתמש מספר טבעי ופולטת שורה המורכבת ממקפים (-), שמספרם כערך המספר שהוזן. כל מקף יופרד ממשנהו על ידי כוכבית (*), כאשר גם המקפים הקיצוניים (בתחילת השורה ובסופה) יימצאו בין שתי כוכביות. למשל, עבור הערך 5, יוצג המלל הבא :

* _ * _ * _ * _ * _ *

הלולאה do-while

4. כתוב לולאות do-while הקולטות מהמשתמש ערכים שלמים ומוודאות כי הם :

1. חיוביים.
2. שונים מ-0.
3. בין 4 ל-15 (כולל).
4. שליליים וגדולים מ-100.
5. בין 0 ל-900 (כולל) ומתחלקים ב-3 ללא שארית.

5. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מספרים חיוביים מהמשתמש שוב ושוב, עד שמוזן מספר הקטן מהמספר שהוזן לפניו.
2. תוכנית הקולטת מספר טבעי מהמשתמש ופולטת את סכום ספרותיו.
3. מכונית נוסעת במהירות קבועה כלשהי. כאשר היא עוברת על פני נקודה המוגדרת כנקודת האפס, משנה הנהג את מהירותה. החל מרגע זה, משנה הנהג את מהירות מכוניתו בסופה של כל שעה עגולה. כעבור מספר שעות מחליט הנהג לעצור את מכוניתו, ומהירותה יורדת לאפס בן-רגע (כלומר, המכונית נעצרת כאשר יוזן הערך 0). כתוב תוכנית הקולטת את מהירות המכונית (בקמ"ש) בכל פעם שהיא משתנה, ופולטת את:
 - ❖ הזמן (בשעות) שנסעה המכונית.
 - ❖ המרחק (בק"מ) שעברה המכונית מנקודת האפס.
 - ❖ המהירות הממוצעת (בקמ"ש) בה נסעה המכונית.

6. לפניך מספר תוכניות.

- א. ציין מהו הפלט המתקבל מכל אחד מהקלטים המצוינים תחתן.
- ב. בהסתמך על סעיף א', נסה לקבוע מה מבצעת כל תוכנית.

1.

בתוכנית זו, ערך המשתנים a ו-b הם טבעיים

```
#include <stdio.h>
void main()
{
    int a, b, c;

    printf("Please enter two numbers: ");
    scanf("%d %d", &a, &b);

    do {
        c = a;
        a = b;
        b = c % b;
    } while (b != 0);

    printf("Result: %d \n", c);
}
```

הפלט עבור a=35, b=7

הפלט עבור a=100, b=90

הפלט עבור a=16, b=12

2.

בתוכנית זו, ערך המשתנה n הוא שלם וגדול מ-1

```
#include <stdio.h>
void main()
{
    int n, a = 1, sum = 0;

    printf("Please enter a number: ");
    scanf("%d", &n);

    printf("0 ");
    do {
        printf("%d ", a);
        sum += a * a;
        a = sum / a;
        n--;
    } while (n > 1);
}
```

הפלט עבור n=5

הפלט עבור n=7

הפלט עבור n=9

3.

בתוכנית זו, ערך המשתנה a הוא אי-שלילי

(הערה: מומלץ להשתמש במחשבון כדי למצוא את פלט התוכנית)

```
#include <stdio.h>
void main()
{
    float a, c = 1;

    printf("Please enter a number: ");
    scanf("%f", &a);

    do
        c = (c + a / c) / 2.0;
    while (a - c * c > 0.0001 || c * c - a > 0.0001);

    printf("Result: %f \n", c);
}
```

הפלט עבור a=9 :

הפלט עבור a=25 :

הפלט עבור a=36 :

4.

בתוכנית זו, ערך המשתנים num1 ו-num2 הם טבעיים

```
#include <stdio.h>
void main()
{
    int num1, num2, q, p, t = 0, n = 1, x = 0;

    printf("Please enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    while (num1 > 0 || num2 > 0 || t > 0) {
        q = num1 % 10;
        p = num2 % 10;
        x += (q + p + t) % 10 * n;
        t = (q + p + t) / 10;
        n *= 10;
        num1 /= 10;
        num2 /= 10;
    }

    printf("Result: %d \n", x);
}
```

הפלט עבור num1=6, num2=4 :

הפלט עבור num1=18, num2=22 :

הפלט עבור num1=19, num2=5 :

הלולאה for

7. אחדים מקטעי התוכניות שלפניך שגויים. סמן אותם והסבר מהן הטעויות. עבור קטעי התוכניות הנכונים, ציין מהו הפלט המתקבל מהם.

1.

```
for (i = 0 , i < 5 , i++)  
    printf("%d ", 10 % i);
```

2.

```
for (max = 10 ; max >= 4 ; max -= 3)  
    printf("%d ", max * max);
```

3.

```
for (num = 4 ; num < 3 ; num--)  
    printf("Hello! \n");
```

4.

```
for l = 'a' ; l < 'e' ; l++  
    printf("%c \n", l);
```

5.

```
for (x = 4.0 ; x < 10.0 ; x += 1.5) {  
    printf("%f \n", x);  
    x++;  
}
```

6.

```
cnt = 7;  
for (x = 0 ; cnt != 21 && x < 7 ; ) {  
    x++;  
    printf("%d %d %d \n", x, cnt, x * cnt);  
    cnt += 7;  
}
```

7.

```
ch = '0';  
for ( ch <= '9' ; ch += 2) {  
    printf("%d ", ch + 1);  
    ch++;  
}
```

8.

```
f1 = -4;  
f2 = 3;  
for (f2 = f1 - f2 ; f1 < 0 ; f1++) {  
    printf("%d \n", f1 * f2);  
    f2--;  
}
```

8. לפניך מספר קטעי תוכניות המכילים לולאות while ו-for. הפוך את לולאות while ללולאות for ואת לולאות for ללולאות while.

1.

```
rate = 0.0;
while (rate < 0.5) {
    printf("rate: %f \n", rate);
    rate += 0.1;
}
```

2.

```
for (pt = 100 ; pt >= 0 ; pt -= 7)
    printf("pt = %d \n", pt);
```

3.

```
while (count < 9.25) {
    count += 2.25;
    printf("%f ", count * 4);
}
```

4.

```
for ( ; x != x * x ; )
    scanf("%f", &x);
```

5.

```
while (num >= max)
    scanf("%d", &num);
```

6.

```
for (i = 0 ; i <= 10 ; i++) {
    scanf("%d", &num);
    sum += num;
}
```

9. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש תו ומספר, ופולטת את התו על המסך מספר פעמים השווה למספר שהוזן.
2. תוכנית הקולטת מהמשתמש שני מספרים חיוביים, כאשר שניהם זוגיים והראשון קטן מהשני, ופולטת את סכום המספרים הזוגיים שביניהם.
3. תוכנית הקולטת מהמשתמש מספר טבעי ופולטת את כל עצרות המספרים הטבעיים עד אליו (השתמש במשתנה מסוג long לחישוב העצרת, כדי שתוכל

לחשב עצרות של מספרים גבוהים יחסית). למשל, עבור המספר 4, יוצג הפלט הבא:

$$\begin{aligned} 1! &= 1 \\ 2! &= 2 \\ 3! &= 6 \\ 4! &= 24 \end{aligned}$$

10. כתוב את התוכניות הבאות:

1. תוכנית הפולטת את סכום n האיברים הראשונים (n הוא מספר טבעי שמזין המשתמש) של הסדרה הבאה:

$$\frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \dots + \frac{n}{n}$$

2. תוכנית הפולטת את סכום n האיברים הראשונים (n הוא מספר טבעי שמזין המשתמש) של הסדרה הבאה (מומלץ להשתמש במשתנה מסוג long לחישוב ערך הסדרה):

$$n^1 - n^2 + n^3 - n^4 + n^5 - \dots + n^n$$

3. תוכנית הפולטת את n האיברים הראשונים של סדרת פיבונאצ'י, כאשר n הוא מספר טבעי גדול מ-1 שמזין המשתמש.

11. כתוב את התוכניות הבאות:

1. תוכנית המשהה את פעולת המחשב למספר שניות שמזין המשתמש. לשם כך השתמש בלולאת for שגופה ריק, ומתבצע פעמים רבות (תוכנית זו תרוץ באופן שונה על מחשבים שונים, ולכן לא מומלץ להשתמש בשיטה זו כדי להשהות את המחשב לפרק זמן מסוים).

2. תוכנית הפולטת את השטח המוגבל על ידי גרף הפונקציה $3x^2$ וציר ה- x , בגבולות a ו- b הנקלטים על ידי המשתמש ($a < b$). עשה זאת על ידי סיכום ערכי הפונקציה עבור ערכי x השואפים לאפס, בתחום a ו- b .

3. תוכנית הקולטת מספר אי-שלילי שלם מהמשתמש ומציגה הודעה מתאימה אם מכפלתו במספר ההפוך לו (בסדר ספרותיו) מתחלקת ב-7 ללא שארית. למשל, הודעה שכזו תוצג עבור המספר 14, משום שמכפלתו במספר ההפוך לו בסדר הספרות, 41 ($41 = 14 * 41$), מתחלקת ב-7 ללא שארית.

אופרטור הפסיק

12. לפניך מספר קטעי קוד המכילים ביטויי פסיק. מצא מהם ערכי המשתנים לאחר ביצוע כל אחד מהם.

1.

```
int result;  
result = 4 * 3, 5 * 2;
```

2.

```
int a, b = 1;  
a = 0, b++;
```

3.

```
int num, a = 0, b = 1;  
num = a, a = b;
```

4.

```
char ch, p = 'n', r = 'f';  
ch = (++p, --r);
```

5.

```
int rad;  
rad = 10,000;
```

6.

```
int deg;  
deg = (10,000);
```

7.

```
char a, b, c;  
a = 'a', b = 'b', c = 'c';
```

8.

```
float m = 0.0, n = 1.5;  
m = m, ++n;  
n = (m++, ++m);
```

13. לפניך מספר לולאות בעלות ביטויי פסיק. מצא מהם ערכי המשתנים לאחר ביצוע כל אחת מהן.

1.

```
float num;
do
    scanf("%f", &num);
while (num != 1.0, num + 1 != 1.0);
```

2.

```
int s, n;
for (s = 0, n = 1 ; n <= 10 ; s += n, n++) {
    /* empty body */
}
```

3.

```
int i;
for (i = 0, --i ; i < 10 ; i++, i++, i++) {
    /* empty body */
}
```

4.

```
char c, p, q;
for (c = 1 , p = 'a', q = 'z' ; p != q ; p++, c++) {
    /* empty body */
}
```

5.

```
int i, q, a, b;
for (i = 2, a = 0, b = 1 ; i <= 6 ; q = b, b += a, a = q, i++) {
    /* empty body */
}
```

6.

```
int s1, s2, i;
for (i = 0, s1 = 0, s2 = 0 ; i < 20 ; s1 += 3, s2 += 5, i++) {
    /* empty body */
}
```

שילוב לולאות ומשפטי תנאי

14. כתוב את התוכניות הבאות (בחר בלולאה המתאימה ביותר):

1. תוכנית הקולטת מהמשתמש מספר טבעי ופולטת הודעה מתאימה אם הוא מספר מושלם. מספר מושלם הוא מספר השווה לסכום כל המספרים (פרט לעצמו) בהם הוא מתחלק ללא שארית. למשל, 6 הוא מספר כזה, משום ש- $1+2+3=6$. גם 28 הוא מספר כזה, משום ש- $1+2+4+7+14=28$.
גורם משותף של שני מספרים הוא מספר בו מתחלקים שני המספרים ללא שארית. כתוב תוכנית הקולטת מהמשתמש שני מספרים טבעיים ופולטת את הגורם המשותף הגדול ביותר שלהם (למשל, הגורם המשותף הגדול ביותר של המספרים 15 ו-25 הוא 5).
3. תוכנית הקולטת מהמשתמש מספר אי-שלילי שלם ופולטת את הספרה הקטנה ביותר ואת הספרה הגדולה ביותר שלו.

15. כתוב את התוכניות הבאות (בחר בלולאה המתאימה ביותר):

1. תוכנית הקולטת מהמשתמש עשרה מספרים ופולטת את המספר הקטן ביותר שהוזן.
2. תוכנית הקולטת מספר שלם גדול מ-1 ופולטת הודעה מתאימה אם הוא מספר ראשוני.
3. π (פאי) הוא קבוע מתמטי שערכו המקורב, בדיוק של שש ספרות אחרי הנקודה העשרונית, הוא 3.141592. ניתן להתקרב לערך של π באמצעות המשוואה הבאה:

$$\frac{\pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \frac{8}{9} \times \dots$$

כתוב תוכנית המחשבת בדרך זו את הערך המקורב של π ופולטת אותו.

לולאות מקוננות

16. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש שני מספרים טבעיים, המייצגים אורך ורוחב של מלבן, ופולטת מלבן מלא מכוכביות, במימדים המתאימים. למשל, עבור הערכים 5 ו-3, יוצג המלבן הבא:

```
*****  
*****  
*****
```

2. תוכנית המציגה על המסך שורת כוכביות, שמספרן כמספר שלם בין 1 ל-20 (כולל) שמזין המשתמש. את הכוכביות יש להציג במעין אנימציה, כלומר, תוך השהייה הולכת וגדלה בין הצגת כל כוכבית וכוכבית.

3. תוכנית הקולטת מספר מהמשתמש ופולטת את כל המספרים העשירים עד אליו. מספר עשיר הוא מספר טבעי הקטן מסכום כל המספרים (פרט לעצמו) בהם הוא מתחלק ללא שארית. למשל, 12 הוא מספר שכזה, משום ש-12 קטן מ- $1+2+3+4+6$.

17. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספר טבעי n ופולטת משולש ישר-זווית מכוכביות, בגובה n שורות. למשל, עבור הערך 5, יוצג המשולש הבא:

```
*****  
****  
***  
**  
*
```

2. תוכנית הקולטת מהמשתמש מספר טבעי אי-זוגי n ופולטת משולש שווה-שוקיים מכוכביות, שבסיסו תחתון ומורכב מ- n כוכביות. למשל, עבור הערך 7, יוצג המשולש הבא:

```
*  
***  
*****  
*****
```

3. תוכנית הקולטת מהמשתמש שני מספרים טבעיים, המייצגים אורך ורוחב של מלבן, ופולטת מלבן ריק מכוכביות, במימדים המתאימים. למשל, עבור הערכים 6 ו-4, יוצג המלבן הבא:

```
*****  
*   *  
*   *  
*****
```

18. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספר טבעי n הקטן מ-10 ופולטת משולש ישר-זווית בן n שורות, כשכל שורה מורכבת ממספרים טבעיים עוקבים החל מ-1. למשל, עבור הערך 4, יוצג המשולש הבא:

```
1
12
123
1234
```

2. תוכנית הקולטת מהמשתמש מספר טבעי n הקטן מ-10 ופולטת משולש ישר-זווית בן n שורות, המורכב מרצפים של המספרים הטבעיים מ-1 עד 9. למשל, עבור הערך 5, יוצג המשולש הבא:

```
1
23
456
7891
23456
```

3. תוכנית הקולטת מהמשתמש מספר טבעי n ופולטת מקבילית בגובה וברוחב n מכוכביות. למשל, עבור הערך 4, תוצג המקבילית הבאה:

```
****
****
****
****
```

19. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספר טבעי ופולטת את כל המספרים הראשוניים הקטנים ממנו בהם הוא מתחלק ללא שארית.

2. תוכנית המציגה את 100 המספרים הטבעיים הראשוניים במשחק 1-2-3 בוס, כאשר "בוס" הוא מספר חד-ספרתי שמזין המשתמש. כלומר, יש לבדוק עבור כל אחד מ-100 המספרים אם הוא מתחלק במספר "בוס" ללא שארית, או אם אחת מספרותיו היא המספר "בוס". אם כן, תוצג המילה "בוס". אם לא, יוצג המספר עצמו.

3. תוכנית הקולטת מהמשתמש מספר טבעי n ופולטת את כל המספרים בעלי n ספרות, השווים לסכום כל אחת מספרותיהם בחזקת n . למשל, אם n הוא 3, יוצג (בין היתר) המספר 153, מכיון שהוא שווה ל- $1^3+5^3+3^3$. אם n הוא 4, יוצג (בין היתר) המספר 1634, מכיון שהוא שווה ל- $1^4+6^4+3^4+4^4$.

20. לפניך מספר תוכניות. ציין מהו הפלט המתקבל מכל אחד מהקלטים המצוינים תחתן. בהסתמך על כך, קבע מה מבצעת כל תוכנית.

1.

בתוכנית זו, ערך המשתנה n הוא טבעי

```
#include <stdio.h>
void main()
{
    int n, x;

    printf("Please enter a number: ");
    scanf("%d", &n);

    while (n > 0) {
        for (x = 1 ; x < n ; x++)
            printf(" ");
        printf("* \n");
        n--;
    }
}
```

הפלט עבור n=3

הפלט עבור n=6

2.

בתוכנית זו, המשתנה n מייצג מספר טבעי בעל ארבע ספרות השונות מאפס

```
#include <stdio.h>
void main()
{
    int a, d = 1000, j, i;

    printf("Please enter a number: ");
    scanf("%d", &a);

    do {
        j = a / d % 10;
        for (i = 0 ; i < j ; i++)
            printf("%d", j);
        printf(" ");
        d /= 10;
    } while (d > 0);
}
```

הפלט עבור a=4132

הפלט עבור a=6231

3.

בתוכנית זו, ערך המשתנה n הוא טבעי ואי-זוגי

```
#include <stdio.h>
void main()
{
    int n, i, q = 0, m, t = 1;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (m = n - 2 ; n > 0 ; n--) {
        for (i = 0 ; i < q ; i++)
            printf(" ");
        printf("*");

        for (i = 0 ; i < m; i++)
            printf(" ");
        if (m > 0)
            printf("*");
        else
            t *= -1;

        q += t;
        m -= t * 2;
        printf("\n");
    }
}
```

הפלט עבור n=3:

הפלט עבור n=5:

לולאות אינסופיות

21. מצא אילו מהלולאות הבאות הן לולאות אינסופיות:

1.

```
while (0 == 0)
    printf("Intruder! Intruder! \n");
```

2.

```
for (p = 10 ; p < 1000 ; p++)
    printf("%d \n", p + p);
```

3.

```
do {
    scanf("%f", &x);
    x++;
} while (4.0 <= 8.0 / 2.0);
```

4.

```
while (len == len - 1)
    printf("len is %d \n", len);
```

5.

```
for (var = 0 ; var > 0 ; var++)
    printf("%d \n", p + p);
```

6.

```
for ( ; i - i >= 0 ; i++)
    printf("i = %c \n", i);
```

7.

```
do
    scanf("%d", &num);
while (num > 0);
```

8.

```
while (a = b ; ; a *= a)
    printf("Is this an infinite loop? \n");
```

משפטי בקרה נוספים

המשפט goto

22. אחדים מקטעי התוכניות שלפניך שגויים. סמן אותם והסבר מהן הטעויות. עבור קטעי התוכניות הנכונים, ציין מהו הפלט המתקבל מהם.

1.

```
printf("kid");
goto end;
printf("ding");
end:
    printf("ney");
```

2.

```
ten:
    printf("*");
    goto ten;
```

3.

```
printf("\n\n");
goto for;
xyz:
    printf("Hello, ");
for:
    printf("world! \n");
```

4.

```
int num = 7;
printf("%d", num);
label1:
    num -= 3;
    if (num > 0)
        goto label1;
    printf("%d", num);
```

5.

```
float var = 1.25;
1:
    printf("%f", var * 2);
2:
    goto 3;
3:
    if (var == 1.25 * 2)
        goto 1;
```

6.

```
int x = 0;
pp:
    for ( ; x < 10 ; x += 3) {
jump:
    printf("%d ", x);
    goto qq;
    }
    goto end;
qq:
    x += 5;
    printf("%d ", x);
    goto pp;
end:
    printf("%d ", x);
```

7.

```
int n, s;
s:
    n = 12345;
n:
    s = 0;
_:
    s++;
    n /= 10;
    if (n > 0)
        goto _;
    printf("%d", s);
```

8.

```
float v, x = 1.0;
input:
    v = 0.0;
initialize:
    v += x;
    if (v <= 100.0)
        goto initialize;
    printf("%f", v / (x + 1));
    goto end;
```

9.

```
int x = 4, y = 10;
while (x <= y) {
    x++;
    if (!(x < 6)) {
        goto otop;
        y--;
    }
}
otop:
    printf("%d", y - x);
```

10.

```
char letter = 'a';
printf("%c", letter);
window:
    letter = '-';
    printf("%c", letter);
    goto door;
floor:
    letter++;
    if (letter < 'e')
        goto table;
```

```

    goto end;
door:
    letter = 'b';
table:
    printf("%c", letter);
    goto floor;
end:
    printf("%c", letter);

```

המשפט break

23. אחדים מקטעי התוכניות שלפניך שגויים. סמן אותם והסבר מהן הטעויות. עבור קטעי התוכניות הנכונים, ציין מהו הפלט המתקבל מהם.

1.

```

while (0 == 0) {
    printf("123");
    if (0 == 0) {
        printf("456");
        break;
    }
    printf("789");
}

```

2.

```

int x = 7;
do {
    printf("%d ", x);
    if (x < 3)
        break;
    x--;
} while (1 != 0);

```

3.

```

char ch = 'q';
for ( ; ; ) {
    ch--;
    if (ch > 'l')
        printf("%c", ch);
    else
        break;
}

```

4.

```
float t = 6.5;
jump:
    t += t;
    printf("%f ", t);
    if (t > 100.0)
        break;
    goto jump;
```

5.

```
int i, j;
for (i = 1 ; i <= 10 ; i++) {
    for (j = 1 ; j <= 10 ; j++) {
        if (j == 4)
            break;
        printf("%d ", i * j);
    }
    if (i == 3)
        break;
    printf("\n");
}
```

6.

```
switch (4) {
case 4:
    printf("Ask me what \n");
case 3:
    printf("What? ");
    if (4 == 4)
        break;
case 5:
    printf("Nothing. \n");
}
```

7.

```
int n;
switch (1) {
case 0:
    printf("HELLO from 0 \n");
case 1:
    printf("HELLO from 1 \n");
    for (n = 33 ; ; n++) {
        printf("%d ", n);
        if (n > 37)
            break;
    }
}
```

```

case 2:
    printf("\nHELLO from 2 \n");
}

```

8.

```

switch (2) {
case 1:
    printf("1");
case 2:
    printf("2");
    switch (3) {
case 3:
    printf("a");
case 2:
    printf("b");
case 1:
    printf("c");
    if (3 == 3)
        break;
    }
case 3:
    printf("3");
}

```

24. כתוב את התוכניות הבאות (השתמש במשפטי break):

1. תוכנית הקולטת מהמשתמש מספרים אי-שליליים ופולטת את סכומם ואת הממוצע שלהם. קליטת המספרים תתבצע עד שיוזנו עשרה ציונים או עד שיוזן ערך שלילי.
2. תוכנית הקולטת מהמשתמש עשרים מספרים ולבסוף פולטת את מכפלתם זה בזה. אם ברגע מסוים מכפלתם תהיה גדולה מ-1000 או קטנה מ-0, תסתיים הלולאה הקולטת את הערכים.
3. בתחילת שנה מופקד בחשבון חיסכון סכום כסף כלשהו. באמצע כל חודש עולה הסכום שבחשבון החיסכון ב-interest אחוזים. בסוף כל חודש בודק הבנק האם הסכום שבחשבון החיסכון גדול פי 4 או יותר מהסכום שהופקד בתחילת השנה. אם כן, מקפיא הבנק את החיסכון הזה עד לסוף השנה, ולא משנה אותו. כתוב תוכנית הקולטת את interest ואת הסכום המופקד בתחילת השנה, ופולטת את הסכום שבחשבון החיסכון כעבור שנה.

המשפט continue

25. אחדים מקטעי התוכניות שלפניך שגויים. סמן אותם והסבר מהן הטעויות. עבור קטעי התוכניות הנכונים, ציין מהו הפלט המתקבל מהם.

1.

```
int n = 0;
while (n < 4) {
    printf("#");
    n++;
    if (!2 != !2) {
        n--;
        printf("?");
    }
    continue;
    printf("!");
}
```

2.

```
float d = 1.0;
do {
    d++;
    if (d < 4.0)
        continue;
    printf("%f", d);
    break;
} while (1 == 1);
```

3.

```
int n = 3;
switch (n) {
case 2:
    printf("2");
    if (n < 3)
        printf(" < 3");
case 3:
    printf("3");
    if (n < 4)
        printf(" < 4");
    continue;
case 4:
    printf("4");
    if (n < 5)
        printf(" < 5");
}
```

4.

```
char ch;
for (ch = 'a' ; ch <= 'j' ; ch++) {
    if (ch == 'a' || ch == 'e' || ch == 'i')
        continue;
    printf("%c", ch + 1)
}
```

5.

```
int x = 6, a = 1;
do {
    a *= x;
    x--;
    printf("%d ", a);
    if (x > 1)
        continue;
    break;
} while (1 == 1);
```

6.

```
int i = 5, n, sum = 1;
while (sum == 1) {
    for (n = i ; n > 0 ; ) {
        sum *= n;
        n--;
    }
    if (i > 0)
        continue;
}
printf("%d %d", sum, i);
```

7.

```
int n = 6, m = 1;
float d = 8.5;
if (n < d)
    if (n < m)
        printf("%d ", n * m);
    else if (n == 6)
        printf("%f ", d * 2);
    else {
        printf("Else!");
        continue;
    }
```

8.

```
char from, to = '9';
for (from = '0' ; from < to ; ) {
    do {
        from++;
        if (!(to < from))
            continue;
    } while (0 == 1);
    printf("%c ", from);
}
```

26. כתוב את התוכניות הבאות (השתמש במשפטי continue):

1. תוכנית הקולטת מהמשתמש מספרים טבעיים עד להזנת המספר 0. התוכנית תמצא אילו מהמספרים הם ערכי ASCII של אחת מהספרות, ותחשב את סכומן של ספרות אלו.
2. תוכנית הקולטת מהמשתמש מספרים ופולטת את סכום המספרים שהוזנו הגדולים מ-50. קליטת הערכים תיפסק לאחר שסכום המספרים יהיה גדול מ-300.
3. תוכנית הקולטת מהמשתמש מספר שלם גדול מ-1 ופולטת את המספר הקטן ביותר והגדול ביותר (פרט למספר לעצמו) בו הוא מתחלק ללא שארית.

תשובות

.1

0 60 10 55 20 50 30 45	.5 ✓	* * *	.1 ✓
-0- -1- -0- -1- -0-	.6 ✓	יש לתחום את ביטוי הלולאה בסוגריים	.2 ✗
3 6 12 24	.7 ✓	hello, world hello, world hello, world	.3 ✓
41322314	.8 ✓	על המילה השמורה while להיכתב באותיות קטנות	.4 ✗

.2

1.

```
#include <stdio.h>
void main()
{
    int num1, num2, counter = 0, total = 0;

    printf("Please enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    while (counter < num1) {
        total += num2;
        counter++;
    }

    printf("%d * %d = %d \n", num1, num2, total);
}
```

2.

```
#include <stdio.h>
void main()
{
    int num1, num2, counter, total = 0;

    printf("Please enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    counter = num1;
```

```

while (counter > 0) {
    counter -= num2;
    total++;
}

printf("%d / %d = %d \n", num1, num2, total);
}

```

3.

```

#include <stdio.h>
void main()
{
    int num1, num2, total = 0;

    printf("Please enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    total = num1;
    while (total - num2 >= 0)
        total -= num2;

    printf("%d %% %d = %d \n", num1, num2, total);
}

```

.3

1.

```

#include <stdio.h>
void main()
{
    char letter1, letter2;

    printf("Please enter two letters: ");
    scanf("%c %c", &letter1, &letter2);

    while (letter1 <= letter2) {
        printf("%c ", letter1);
        letter1++;
    }

    printf("\n");
}

```

2.

```
#include <stdio.h>
void main()
{
    float e = 1;
    int n = 100;      /* also try n = 1000, n = 10000 */
    int counter = 0;

    while (counter < n) {
        e *= (1.0 + 1.0 / n);
        counter++;
    }

    printf("e is: %f \n", e);
}
```

3.

```
#include <stdio.h>
void main()
{
    int num, counter = 0;

    printf("Please enter a number: ");
    scanf("%d", &num);

    while (counter < num) {
        printf("*-");
        counter++;
    }

    printf("* \n");
}
```

.4

1.

```
do
    scanf("%d", &num);
while (num <= 0);
```

2.

```
do
    scanf("%d", &num);
while (num == 0);
```

3.

```
do
    scanf("%d", &num);
while (num < 4 || num > 15);
```

4.

```
do
    scanf("%d", &num);
while (num >= 0 || num <= -100);
```

5.

```
do
    scanf("%d", &num);
while (num < 0 || num > 900 || num % 3 != 0);
```

.5

1.

```
#include <stdio.h>
void main()
{
    float num = 0, last;

    do {
        last = num;
        printf("Please enter a number: ");
        scanf("%f", &num);
    } while (num >= last);
}
```

2.

```
#include <stdio.h>
void main()
{
    int num, dig_sum = 0;

    printf("Please enter a number: ");
    scanf("%d", &num);

    do {
        dig_sum += num % 10;
        num /= 10;
    } while (num > 0);

    printf("Sum of digits is: %d \n", dig_sum);
}
```

3.

```
#include <stdio.h>
void main()
{
    int hours = 0, dist = 0, speed, avg_speed = 0;

    do {
        scanf("%d", &speed);
        dist += speed;
        avg_speed += speed;
        hours++;
    } while (speed != 0);

    hours--;
    printf("Duration of driving: %d hours. \n", hours);
    printf("Total distance: %d km. \n", dist);
    printf("Average speed: %d km/h. \n", avg_speed / hours);
}
```

.6

1.

Result: 7	עבור a=35, b=7
Result: 10	עבור a=100, b=90
Result: 4	עבור a=16, b=12

התוכנית מוצאת ופולטת את הגורם המשותף הגדול ביותר של שני המספרים (כלומר, המספר הגדול ביותר ששני המספרים מתחלקים בו ללא שארית).

2.

0 1 1 2 3	עבור n=5
0 1 1 2 3 5 8	עבור n=7
0 1 1 2 3 5 8 13 21	עבור n=9

התוכנית מציגה את n האיברים הראשונים של סדרת פיבונאצ'י, בהתבסס על תכונת הסדרה שמכפלת איבר נתון באיבר הקודם לו שווה לסכום ריבועי כל האיברים הקודמים לו.

3.

Result: 3.000000	עבור a=9
Result: 5.000000	עבור a=25
Result: 6.000000	עבור a=36

התוכנית מוצאת ופולטת את השורש הריבועי (המקורב) של a.

4.

Result: 10	עבור num1=6, num2=4
Result: 40	עבור num1=18, num2=22
Result: 24	עבור num1=19, num2=5

התוכנית מוצאת ופולטת את סכום שני המספרים, על ידי חיבור בנפרד של כל אחת מספרותיהם.

.7

4.000000	.5 ✓	יש להפריד את ביטויי לולאת	.1 ✗
6.500000		ה-for בתווי נקודה-פסיק	
9.000000			
1 7 7	.6 ✓	100 49 16	.2 ✓
2 14 28			
בתוך הסוגריים שלאחר המילה	.7 ✗	(אין פלט)	.3 ✓
for חייבים להופיע שני תווי			
נקודה-פסיק			
28	.8 ✓	את ביטויי לולאת for יש לתחום	.4 ✗
24		בתווי נקודה-פסיק	
18			
10			

.8

1.

```
for (rate = 0.0 ; rate < 0.5 ; rate += 0.1)
    printf("rate: %f \n", rate);
```

2.

```
pt = 100;
while (pt >= 0) {
    printf("pt = %d \n", pt);
    pt -= 7;
}
```

3.

```
for ( ; count < 9.25 ; ) {
    count += 2.25;
    printf("%f ", count * 4);
}
```

4.

```
while (x != x * x)
    scanf("%f", &x);
```

5.

```
for ( ; num >= max ; )
    scanf("%d", &num);
```

6.

```
i = 0;
while (i <= 10) {
    scanf("%d", &num);
    sum += num;
    i++;
}
```

.9

1.

```
#include <stdio.h>
void main()
{
    int num, i;
    char ch;

    printf("Please enter a number and a character: ");
    scanf("%d %c", &num, &ch);

    for (i = 0 ; i < num ; i++)
        printf("%c", ch);
}
```

2.

```
#include <stdio.h>
void main()
{
    int from, to, sum = 0, i;

    printf("Please enter two numbers: ");
    scanf("%d %d", &from, &to);

    for (i = from ; i <= to ; i += 2)
        sum += i;

    printf("Sum of even numbers from %d to %d: %d \n",
        from, to, sum);
}
```

3.

```
#include <stdio.h>
void main()
{
    int n, i;
    long fact = 1;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (i = 1 ; i <= n ; i++) {
        fact *= i;
        printf("%d! = %ld \n", i, fact);
    }
}
```

.10

1.

```
#include <stdio.h>
void main()
{
    float i, sum = 0;
    int n;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (i = 1.0 ; i <= n ; i++)
        sum += n / i;

    printf("The sum is: %f \n", sum);
}
```

2.

```
#include <stdio.h>
void main()
{
    int i, n, sign = 1;
    long sum = 0, temp = 1;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (i = 1 ; i <= n ; i++) {
        temp *= n;
    }
}
```

```

        sum += temp * sign;
        sign *= -1;
    }

    printf("The sum is: %ld \n", sum);
}

```

3.

```

#include <stdio.h>
void main()
{
    int a1 = 0, a2 = 1, i, n, temp;

    printf("Please enter a number: ");
    scanf("%d", &n);

    printf("%d ", a1);
    for (i = 1 ; i < n ; i++) {
        printf("%d ", a2);
        temp = a2;
        a2 += a1;
        a1 = temp;
    }
}

```

.11

1.

```

#include <stdio.h>
void main()
{
    double sec, d;

    printf("Please enter number of seconds: ");
    scanf("%lf", &sec);

    /* the constant values below might change
     * from one computer to another
     */
    for (d = 20000 ; d > 0 ; d -= 0.0005) {
        /* empty body */
    }
}

```

2.

```
#include <stdio.h>
void main()
{
    float a, b, integral = 0;

    printf("Enter the function limits: ");
    scanf("%f %f", &a, &b);

    for ( ; a <= b ; a += 0.001)
        integral += 3 * a * a * 0.001;

    printf("The integral is: %f \n", integral);
}
```

3.

```
#include <stdio.h>
void main()
{
    int num, same, digit, reverse = 0;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (same = num ; same > 0 ; same /= 10) {
        digit = same % 10;
        reverse = reverse * 10 + digit;
    }

    if (num * reverse % 7 == 0)
        printf("%d * %d divides by 7 with no remainder! \n",
            num, reverse);
}
```

.12

rad: 10	.5	result: 12	.1
deg: 0	.6	a: 0, b: 2	.2
a: 'a', b: 'b', c: 'c'	.7	num: 0, a: 1, b: 1	.3
m: 2.0, n: 2.0	.8	ch: 'e', p: 'o', r: 'e'	.4

c: 26, p: 'z', q: 'z'	.4	num: 0.0	.1
i: 7, q: 5, a: 5, b: 8	.5	s: 55, n: 11	.2
s1: 60, s2: 100, i: 20	.6	i: 11	.3

1.

```
#include <stdio.h>
void main()
{
    int num, div, div_sum = 0;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (div = 1 ; div <= num / 2 ; div++)
        if (num % div == 0)
            div_sum += div;

    if (div_sum == num)
        printf("%d is a perfect number! \n", num);
}
```

2.

```
#include <stdio.h>
void main()
{
    int num1, num2, div, max_div = 1;

    printf("Please enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    for (div = 2 ; div <= num1 && div <= num2 ; div++)
        if (num1 % div == 0 && num2 % div == 0)
            max_div = div;

    printf("Greatest common factor: %d \n", max_div);
}
```

3.

```
#include <stdio.h>
void main()
{
    int num, digit, min_digit = 9, max_digit = 0;

    printf("Please enter a number: ");
    scanf("%d", &num);

    do {
        digit = num % 10;
        if (min_digit > digit)
            min_digit = digit;
        if (max_digit < digit)
            max_digit = digit;
        num /= 10;
    } while (num > 0);

    printf("Min. digit: %d \n", min_digit);
    printf("Max. digit: %d \n", max_digit);
}
```

.15

1.

```
#include <stdio.h>
void main()
{
    float num, min;
    int i;

    printf("Please enter 10 numbers: ");

    for (i = 0 ; i < 10 ; i++) {
        scanf("%f", &num);
        if (i == 0 || num < min)
            min = num;
    }

    printf("Smallest number: %f \n", min);
}
```

2.

```
#include <stdio.h>
void main()
{
```

```

int num, is_prime = 1, i;

printf("Please enter a number: ");
scanf("%d", &num);

for (i = 2 ; i <= num / 2 && is_prime == 1 ; i++)
    if (num % i == 0)
        is_prime = 0;

if (is_prime == 1)
    printf("%d is prime. \n", num);
else
    printf("%d isn't prime. \n", num);
}

```

3.

```

#include <stdio.h>
void main()
{
    float pi = 1, mone = 2, mechane = 1;
    int n;

    for (n = 0 ; n < 10000 ; n++) {
        pi *= mone / mechane;
        if (n % 2 == 1)
            mone += 2;
        else
            mechane += 2;
    }
    pi *= 2;

    printf("Pi is: %f \n", pi);
}

```

.16

1.

```

include <stdio.h>
void main()
{
    int x, y, i;

    printf("Please enter length and width: ");
    scanf("%d %d", &x, &y);
}

```

```

for ( ; y > 0 ; y--) {
    for (i = 0 ; i < x ; i++)
        printf("*");
    printf("\n");
}
}

```

2.

```

#include <stdio.h>
void main()
{
    float time = 15000, cnt;
    int num;

    printf("Enter number of asterisks: ");
    scanf("%d", &num);

    for ( ; num > 0 ; num--) {
        printf("*");
        for (cnt = time ; cnt > 0 ; cnt -= 0.0015) {
            /* body is empty */
        }
        time /= 1.1;
    }
}

```

3.

```

#include <stdio.h>
void main()
{
    int i, num, div_sum, div;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (i = 1 ; i <= num ; i++) {
        div_sum = 0;

        for (div = 1 ; div <= i / 2 ; div++)
            if (i % div == 0)
                div_sum += div;

        if (div_sum > i)
            printf("%d \n", i);
    }
}

```

1.

```
#include <stdio.h>
void main()
{
    int num, i;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for ( ; num > 0 ; num--) {
        for (i = 0 ; i < num ; i++)
            printf("*");
        printf("\n");
    }
}
```

2.

```
#include <stdio.h>
void main()
{
    int num, i, x;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (x = 1 ; x <= num ; x += 2) {
        for (i = 0 ; i < (num - x) / 2 ; i++)
            printf(" ");
        for (i = 0 ; i < x ; i++)
            printf("*");
        printf("\n");
    }
}
```

3.

```
#include <stdio.h>
void main()
{
    int length, width, x, y;
    char ch;

    printf("Please enter length and width: ");
    scanf("%d %d", &length, &width);
```

```

for (y = 1 ; y <= width; y++) {
    printf("*");

    if (y == 1 || y == width)
        ch = '*';
    else
        ch = ' ';
    for (x = 2 ; x <= length - 1; x++)
        printf("%c", ch);

    printf("* \n");
}
}

```

.18

1.

```

#include <stdio.h>
void main()
{
    int n, y, x;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (y = 1 ; y <= n ; y++) {
        for (x = 1 ; x <= y ; x++)
            printf("%d", x);
        printf("\n");
    }
}

```

2.

```

#include <stdio.h>
void main()
{
    int n, y, x, num = 0;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (y = 1 ; y <= n ; y++) {
        for (x = 1 ; x <= y ; x++) {
            printf("%d", num % 9 + 1);
            num++;
        }
    }
}

```

```
    printf("\n");
}
}
```

3.

```
#include <stdio.h>
void main()
{
    int n, x, y;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (y = 1 ; y <= n ; y++) {
        for (x = 1 ; x < y ; x++)
            printf(" ");
        for (x = 1 ; x <= n ; x++)
            printf("*");
        printf("\n");
    }
}
```

.19

7.

```
#include <stdio.h>
void main()
{
    int num, i;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (i = 2 ; i <= num / 2 ; i++) {
        if (num % i == 0) {
            printf("%d \n", i);

            do
                num /= i;
            while (num % i == 0);
        }
    }
}
```

2.

```
#include <stdio.h>
void main()
{
    int boom, n, i, flag;

    printf("Please enter the \"boom\" number: ");
    scanf("%d", &boom);

    for (i = 1 ; i < 100 ; i++) {
        flag = 1;

        if (i % boom == 0)
            flag = 0;

        for (n = i ; flag == 1 && n > 0 ; n /= 10)
            if (n % 10 == boom)
                flag = 0;

        if (flag == 1)
            printf("%d \t", i);
        else
            printf("boom! \t");
    }
}
```

3.

```
#include <stdio.h>
void main()
{
    int n, mult, from = 1, to = 10, i, sum, temp;

    printf("Please enter the value 'n': ");
    scanf("%d", &n);

    for (i = 1 ; i < n ; i++) {
        from *= 10;
        to *= 10;
    }

    for ( ; from < to ; from++) {
        temp = from;
        sum = 0;

        do {
            mult = 1;
```

```

        for (i = 0 ; i < n; i++)
            mult *= temp % 10;
        sum += mult;
        temp /= 10;
    } while (temp > 0);

    if (sum == from)
        printf("%d \n", sum);
}
}

```

.20

1.

```

*                                     עבור n=3
*
*
*      *                               עבור n=6
*      *
*      *
*      *
*      *
*
*

```

התוכנית מציגה קו נטוי כלפי מעלה המורכב מ-n כוכביות.

2.

```

4444 1 333 22                       עבור a=4132
666666 22 333 1                      עבור a=6231

```

התוכנית פולטת כל אחת מספרות המספר מספר פעמים השווה לה.

3.

```

* *                                     עבור n=3
*
* *
*      *                               עבור n=5
*      *
*      *
*      *
*      *

```

התוכנית פולטת צורת X שגובהה n כוכביות.

.21

	.5	הלולאה אינסופית	.1
הלולאה אינסופית	.6		.2
	.7	הלולאה אינסופית	.3
הלולאה אינסופית	.8		.4

.22

0 5 5 10 10	.6 ✓	kidney	.1 ✓
5	.7 ✓	(פלט אינסופי של '*' ... *****	.2 ✓
אין לדלג לתווית שלא הוגדרה (end)	.8 ✗	אין לקרוא לתווית בשם של מילה (for)	.3 ✗
4	.9 ✓	7-2	.4 ✓
a-bcde	.10 ✓	אין להתחיל שם של תווית בספרה	.5 ✗

.23

1 2 3 2 4 6 3 6 9	.5 ✓	123456	.1 ✓
Ask me what What?	.6 ✓	7 6 5 4 3 2	.2 ✓
HELLO from 1 33 34 35 36 37 38 HELLO from 2	.7 ✓	ponm	.3 ✓
2abc3	.8 ✓	אין לבצע משפט break מחוץ ללולאות או למשפט switch	.4 ✗

.24

1.

```
#include <stdio.h>
void main()
{
    float sum = 0, num;
    int i;

    printf("Please enter numbers: ");
    for (i = 0 ; i < 10 ; i++) {
        scanf("%f", &num);
```

```

        if (num < 0.0)
            break;
        sum += num;
    }

    printf("Sum of numbers: %f \n", sum);
    printf("Average of numbers: %f \n", sum / i);
}

```

2.

```

#include <stdio.h>
void main()
{
    float num, mult = 1;
    int i;

    printf("Please enter twenty numbers: ");
    for (i = 0 ; i < 20 ; i++) {
        scanf("%f", &num);
        mult *= num;
        if (mult < 0.0 || mult > 1000.0)
            break;
    }

    printf("The multiplication of the numbers is: %f \n",
        mult);
}

```

3.

```

#include <stdio.h>
void main()
{
    float saving, first_saving, interest;
    int months;

    printf("Please enter the saving and the interest: ");
    scanf("%f %f", &saving, &interest);
    first_saving = saving;

    for (months = 0 ; months < 12 ; months++) {
        saving += saving * interest / 100.0;
        if (saving > first_saving * 4.0)
            break;
    }
    printf("The saving after a year: %f \n", saving);
}

```

6 30 120 360 720	.5 ✓	####	.1 ✓
120 5	.6 ✓	4.000000	.2 ✓
אין לבצע משפט continue מחוץ ללולאות	.7 ✗	אין לבצע משפט continue מחוץ ללולאות	.3 ✗
1 2 3 4 5 6 7 8 9	.8 ✓	cdeghik	.4 ✓

1.

```
#include <stdio.h>
void main()
{
    int num, sum = 0;

    printf("Please enter numbers: ");
    do {
        scanf("%d", &num);

        if (num < '0' || num > '9')
            continue;
        sum += num - '0';
    } while (num != 0);

    printf("Sum of digits: %d \n", sum);
}
```

2.

```
#include <stdio.h>
void main()
{
    float num, sum = 0;

    printf("Please enter numbers: ");
    while (sum <= 300.0) {
        scanf("%f", &num);
        if (num <= 50.0)
            continue;
        sum += num;
    }

    printf("The sum is: %f \n", sum);
}
```

3.

```
#include <stdio.h>
void main()
{
    int num, i, min_div = 1, max_div = 1;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (i = 2; i <= num / 2 ; i++) {
        if (num % i != 0)
            continue;
        if (i > max_div)
            max_div = i;
        if (min_div == 1)
            min_div = i;
    }

    printf("Min. divider: %d \n", min_div);
    printf("Max. divider: %d \n", max_div);
}
```

פרק 8

פונקציות

שים לב!

בפרק זה מופיעות פונקציות קצרות ביותר, הכוללות הוראה או שתיים בלבד. הדבר נעשה כדי להקל על הבנת תרגילים בהם יש לעמוד על נקודה מסוימת. חשוב לזכור, שבכתיבת תוכניות מלאות ושימושיות אין לנהוג כך.



שאלות

הגדרת הפונקציה

1. אחדות מהגדרות הפונקציות שלפניך שגויות. סמן אותן והסבר מהן הטעויות.

```
1.  
void Func()  
{  
    /* ... */  
}
```

```
2.  
int Display  
{  
    /* ... */  
}
```

3.

```
float Concrete()
{
    /* ... */
}
```

4.

```
void Calculate();
{
    /* ... */
}
```

5.

```
int Queue()
```

6.

```
float Routine()
{
    /* ... */
}
```

פונקציות פשוטות

2. כתוב פונקציה בשם PrintNum הקולטת מספר מהמשתמש ופולטת אותו חמש פעמים.

3. הסבר מדוע לא ניתן לקרוא לפונקציה PrintNum שכתבת בתרגיל הקודם באופנים הבאים:

```
1. PrintNum;
2. PrintNum(10);
3. i = PrintNum();
4. printnum();
5. void PrintNum();
```

4. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספר טבעי n ופולטת מלבן בעל n שורות, כשכל שורה מורכבת מ-7 כוכביות. הפונקציה main תקרא לפונקציה בשם PrintLine, שתדפיס שורה בודדת של כוכביות.

2. תוכנית הקולטת סדרות בנות עשרה מספרים שלמים ופולטת את הממוצע של כל אחת לאחר קליטתה. הפונקציה main תקרא לפונקציה בשם ReadNumbers, שתקלוט מהמשתמש עשרה מספרים ותפלוט את ממוצעם. לאחר כל קליטת סדרה ופליטת ממוצע מספריה, יישאל המשתמש אם ברצונו להזין סדרת מספרים נוספת או לאו.

3. כתוב את תוכנית 1, כך שלאחר פליטת שורת כוכביות בודדת (לפני שתיפלט השורה הבאה) יתבקש המשתמש ללחוץ על מקש הרווח (' '). הפונקציה main תקרא לפונקציה בשם PauseOutput במקום המתאים, שתסתיים כאשר ילחץ מקש הרווח.

סדר ביצוע פונקציות והחזרה מהן

5. לפניך מספר תוכניות המכילות פונקציות רבות. מצא מהו הפלט המתקבל מהרצת כל תוכנית.

1.

```
#include <stdio.h>

void Call1()
{
    printf("k");
}

void Call2()
{
    printf("Bac");
    Call1();
}

void Call3()
{
    Call1();
    Call2();
}

void Call()
{
    printf("\nTas");
    Call3();
}

void main()
{
    Call();
}
```

2.

```
#include <stdio.h>

void Func1()
{
    printf(" ");
}

void NewLine()
{
    printf("\n");
}

void Func2()
{
    printf(",");
    Func1();
}

void Func3()
{
    printf("world");
}

void Func4()
{
    printf("hello");
    Func2();
}

void Func()
{
    Func4();
    Func3();
}

void main()
{
    NewLine();
    Func();
    NewLine();
}
```

3.

```
#include <stdio.h>

void Sub1()
{
    printf("*****\n");
    Sub1();
}

void Sub2()
{
    printf("\n");
    Sub1();
}

void main()
{
    Sub2();
}
```

6. לפניך תוכנית האמורה לקלוט תווים מהמשתמש עד ללחיצה על מקש הרווח, ולהציג כוכביות במקום את התווים המוזנים. מהי השגיאה בתוכנית?

```
#include <stdio.h>
#include <conio.h>

void ReadChars()
{
    while (getch() != ' ')
        PrintAsterisk();
}

void PrintAsterisk()
{
    printf("*");
}

void main()
{
    ReadChars();
}
```

פונקציות שמקבלות פרמטרים

7. אחדות מהגדרות הפונקציות שלפניך שגויות. סמן אותן והסבר מהן הטעויות.

1.

```
void PrintInt(int i)
{
    /* ... */
}
```

2.

```
void FindFloat(void f)
{
    /* ... */
}
```

3.

```
void ShowNumerals(float num1, int num2, char ch1, char ch2)
{
    /* ... */
}
```

4.

```
void QueueNumbers(int a, b, c)
{
    /* ... */
}
```

5.

```
void DisplayNumbers(int a; int b; int c)
{
    /* ... */
}
```

6.

```
void PrintRange(char ch1, char ch2)
{
    /* ... */
}
```

7.

```
void Max(float x, float y, float z = 100)
{
    /* ... */
}
```

8.

```
void Min(int num1, int num2, int num3)
{
    int num1;
    /* ... */
}
```

8. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספר טבעי ופולטת את סכום המספרים הזוגיים החיוביים הקטנים ממנו, והאם המספר הוא מספר ראשוני. הפונקציה main תקרא לפונקציה בשם PrintSum, שתפלוט את סכום המספרים הזוגיים החיוביים הקטנים מהמספר שנשלח אליה, ולפונקציה בשם PrintPrime, שתפלוט האם המספר שנשלח אליה הוא ראשוני או לא.
2. תוכנית הקולטת מהמשתמש אורך ורוחב של מלבן, ופולטת מלבן מלא ומלבן ריק במימדים שהוזנו. הפונקציה main תקרא לפונקציות PrintFullRect ו-PrintEmptyRect, שיפלטו את המלבן המלא ואת המלבן הריק, בהתאמה.
3. תוכנית הקולטת מהמשתמש סדרות בנות שלושה מספרים, ופולטת עבור כל סדרה את סכום ספרות המספרים שלה. קליטת הסדרות תסתיים כאשר יוזנו שלושה מספרים זהים. הפונקציה main תקרא לפונקציה בשם PrintDigitsSum, שתקבל כפרמטרים שלושה מספרים ותפלוט את סכום ספרותיהם.

פונקציות שמחזירות ערכים

9. כתוב את הפונקציות הבאות:

1. פונקציה בשם GetDigits, המקבלת כפרמטר מספר שלם ומחזירה את סכום ספרותיו.
2. פונקציה בשם GetBiggest, המקבלת כפרמטרים ארבעה מספרים עשרוניים ומחזירה את המספר הגדול מביניהם.
3. פונקציה בשם GetCharType, המקבלת כפרמטר תו ומחזירה את הערך 1 אם הוא ספרה, את הערך 2 אם הוא אות לועזית גדולה, את הערך 3 אם הוא אות לועזית קטנה, ואת הערך 0 במקרה אחר.

1. תוכנית הקולטת מהמשתמש שלושה מספרים, ופולטת כמה מספרים הגדולים מהמספר הראשון, קטנים מהמספר השני ומתחלקים במספר השלישי ללא שארית. הפונקציה main תקרא לפונקציה בשם GetDiv, שתקבל את שלושת המספרים ותחזיר את הערך שיש לפלוט.
2. תוכנית הקולטת מהמשתמש עשרה מספרים, ופולטת כמה ספרות מהם הן אי-זוגיות. הפונקציה main תקרא לפונקציה בשם GetDigit, המקבלת כפרמטר מספר בודד ומחזירה כמה מספרותיו הן אי-זוגיות.
3. תוכנית הקולטת מהמשתמש תווים עד להזנת תו הרווח, ולאחר כל הזנת תו תפלוט אותו אם הוא אות קטנה או גדולה. הפונקציה main תקרא לפונקציה בשם IsCharLetter, המקבלת כפרמטר תו ומחזירה את הערך 1 אם הוא אות קטנה או גדולה, ואת הערך 0 אם לאו.

שימושים נוספים במשפט return

11. לפניך מספר פונקציות המחזירות ערכים. חלק מהפונקציות משתמשות במשפטי return שגויים ; סמן אותן והסבר את הטעויות. עבור הפונקציות המשתמשות במשפטי return נכונים, ציין מהו הערך המוחזר מהן.

```
1.  
int GetDown(float q)  
{  
    return q / 2;  
}
```

```
2.  
int Add(int x)  
{  
    return x++;  
}
```

```
3.  
void Print()  
{  
    printf("Hello! \n");  
    return 0;  
}
```

4.

```
int GetState(int num)
{
    if (num > 0)
        return 1;
    if (num == 0)
        return 0;
    return -1;
}
```

5.

```
float Fraction(float f)
{
    printf("f / 2: %f \n", f / 2);
    return;
}
```

6.

```
int Sub(char ch)
{
    printf("%c %c %c \n", ch, ch + 1, ch + 2);
}
```

הצהרת הפונקציה

12. לפניך שני טורים. הטור הימני מכיל הגדרות פונקציות; הטור השמאלי מכיל הצהרות פונקציות, חלקן נכונות וחלקן שגויות. התאם בין הצהרות הפונקציות הנכונות להגדרות הפונקציות המתאימות להן. הסבר מהן הטעויות בהצהרות הפונקציות השגויות.

- | | | |
|--------------------------------|----------------------------|----|
| 1. float GetMath(char ch); | void Func() | .1 |
| | { | |
| | /* ... */ | |
| | } | |
| 2. float GetMath; | char PressedKey() | .2 |
| | { | |
| | /* ... */ | |
| | } | |
| 3. void Func(); | float GetMath(char ch) | .3 |
| | { | |
| | /* ... */ | |
| | } | |
| 4. char Message(int i); | int f(int i, int j, int k) | .4 |
| | { | |
| | /* ... */ | |
| | } | |
| 5. char Message(int i) | char Message(int i) | .5 |
| | { | |
| | /* ... */ | |
| | } | |
| 6. char PressedKey(); | | |
| | | |
| 7. int f(int i, int j, int k); | | |

13. כתוב את התוכניות מתרגיל 10 תוך שימוש בהצהרות פונקציות. שים לב כי עתה ניתן לשנות את סדר הגדרת הפונקציות בכל תוכנית.

תשובות

.1

אין לרשום נקודה-פסיק לאחר הסוגריים המופיעים לאחר שם הפונקציה	.4 x	.1 ✓
יש להגדיר את גוף הפונקציה על ידי סוגריים מסולסלים פותחים וסוגרים	.5 x	.2 x לאחר שם הפונקציה יש להוסיף סוגריים פותחים וסוגרים
	.6 ✓	.3 ✓

.2

```
/* note: this is not a comelete program,
 * because no main() function is defined.
 */
#include <stdio.h>
void PrintNum()
{
    int i, num;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (i = 0 ; i < 5 ; i++)
        printf("%d \n", num);
}
```

.3

יש לרשום סוגריים פותחים וסוגרים לאחר שם הפונקציה.	.1
אין לרשום דבר בין הסוגריים שלאחר שם הפונקציה.	.2
אין לנסות לקבל ערך מפונקציה שאינה מחזירה ערך.	.3
יש לרשום את שם הפונקציה תוך הבחנה בין אותיות קטנות לגדולות.	.4
אין לרשום את המילה השמורה void בעת הקריאה לפונקציה.	.5

1.

```

#include <stdio.h>

void PrintLine()
{
    printf("***** \n");
}

void main()
{
    int n;

    printf("Please enter n: ");
    scanf("%d", &n);

    while (n > 0) {
        PrintLine();
        n--;
    }
}

```

2.

```

#include <stdio.h>
#include <conio.h>

void ReadNumbers()
{
    int i, num;
    float sum = 0.0;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++) {
        scanf("%d", &num);
        sum += num;
    }

    sum /= 10.0;
    printf("Average of numbers: %f \n", sum);
}

void main()
{
    char ch;

```

```

do {
    ReadNumbers();
    printf("If you want to enter more numbers, press 'y': ");
    ch = getch();
    printf("\n");
} while (ch == 'y' || ch == 'Y');
}

```

3.

```

#include <stdio.h>
#include <conio.h>

void PauseOutput()
{
    while (getch() != ' ') {
        /* empty body */
    }
}

void PrintLine()
{
    printf("***** \n");
}

void main()
{
    int n;

    printf("Please enter n: ");
    scanf("%d", &n);

    while (n > 0) {
        PrintLine();
        PauseOutput();
        n--;
    }
}

```

.5

1.	Taskback
2.	hello, world
3.	***** ***** *****

(פלט אינסופי של שורות כוכביות)

.6

השגיאה –

יש להגדיר את הפונקציה PrintAsterisk לפני שמוגדרת הפונקציה ReadChars.

.7

אין להפריד את הפרמטרים בנקודה-פסיק	.5 X	.1 ✓
	.6 ✓	.2 X אין להגדיר פרמטר מסוג void
אין לאתחל פרמטר בשורת הגדרתו	.7 X	.3 ✓
אין להגדיר משתנה בתוך פונקציה ששמו זהה לשם של פרמטר שמקבלת הפונקציה	.8 X	.4 X יש לציין במפורש את סוגו של כל פרמטר

.8

1.

```
#include <stdio.h>

void PrintSum(int num)
{
    int i, sum = 0;

    for (i = 2 ; i < num ; i += 2)
        sum += i;
    printf("Sum: %d \n", sum);
}

void PrintPrime(int num)
{
    int i, is_prime = 1;
```

```

    if (num == 1)
        is_prime = 0;

    for (i = 2 ; i <= num / 2 ; i++)
        if (num % i == 0)
            is_prime = 0;

    if (is_prime == 1)
        printf("Prime! \n");
    else
        printf("Not prime! \n");
}

void main()
{
    int num;

    printf("Please enter a number: ");
    scanf("%d", &num);

    PrintSum(num);
    PrintPrime(num);
}

```

2.

```

#include <stdio.h>

void PrintFullRect(int x, int y)
{
    int i, j;

    for (i = 0 ; i < y ; i++) {
        for (j = 0 ; j < x ; j++)
            printf("*");
        printf("\n");
    }
}

void PrintEmptyRect(int x, int y)
{
    int i, j;
    char ch;

    for (i = 0 ; i < y ; i++) {
        printf("*");
    }
}

```

```

        ch = ' ';
        if (i == 0 || i == y - 1)
            ch = '*';
        for (j = 1 ; j < x - 1 ; j++)
            printf("%c", ch);

        printf("* \n");
    }
}

void main()
{
    int length, width;

    printf("Please enter length and width: ");
    scanf("%d %d", &length, &width);

    PrintFullRect(length, width);
    printf("\n\n");
    PrintEmptyRect(length, width);
}

```

3.

```

#include <stdio.h>

void PrintDigitsSum(int num1, int num2, int num3)
{
    int sum = 0;

    while (num1 > 0) {
        sum += num1 % 10;
        num1 /= 10;
    }
    while (num2 > 0) {
        sum += num2 % 10;
        num2 /= 10;
    }
    while (num3 > 0) {
        sum += num3 % 10;
        num3 /= 10;
    }

    printf("Sum of digits: %d \n", sum);
}

```

```

void main()
{
    int a, b, c;

    do {
        printf("\nPlease enter three numbers: ");
        scanf("%d %d %d", &a, &b, &c);

        PrintDigitsSum(a, b, c);
    } while (a != b || b != c || a != c);
}

```

.9

1.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
int GetDigits(int num)
{
    int sum = 0;

    while (num > 0) {
        sum += num % 10;
        num /= 10;
    }

    return sum;
}

```

2.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
float GetBiggest(float a, float b, float c, float d)
{
    int max = a;

    if (max < b)
        max = b;
    if (max < c)
        max = c;
    if (max < d)
        max = d;

    return max;
}

```

3.

```
/* note: this is not a complete program,  
 * because no main() function is defined.  
 */  
int GetCharType(char ch)  
{  
    if (ch >= '0' && ch <= '9')  
        return 1;  
    if (ch >= 'A' && ch <= 'Z')  
        return 2;  
    if (ch >= 'a' && ch <= 'z')  
        return 3;  
    return 0;  
}
```

.10

1.

```
#include <stdio.h>  
  
int GetDiv(int a, int b, int c)  
{  
    int i, count = 0;  
  
    for (i = a + 1 ; i < b ; i++)  
        if (i % c == 0)  
            count++;  
  
    return count;  
}  
  
void main()  
{  
    int num1, num2, num3, n;  
  
    printf("Please enter three numbers: ");  
    scanf("%d %d %d", &num1, &num2, &num3);  
  
    n = GetDiv(num1, num2, num3);  
    printf("Result: %d \n", n);  
}
```

2.

```
#include <stdio.h>  
  
int GetDigit(int x)
```

```

{
    int digit, count = 0;

    while (x > 0) {
        digit = x % 10;
        if (digit % 2 == 1)
            count++;
        x /= 10;
    }

    return count;
}

void main()
{
    int i, n = 0, num;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++) {
        scanf("%d", &num);
        n += GetDigit(num);
    }

    printf("%d digits are odd. \n", n);
}

```

3.

```

#include <conio.h>

int IsCharLetter(char ch)
{
    if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')
        return 1;
    return 0;
}

void main()
{
    char ch;

    do {
        ch = getch();
        if (IsCharLetter(ch) == 1)
            putchar(ch);
    } while (ch != ' ');
}

```

.11

✓ 4. הפונקציה מחזירה : 1 אם הערך שנשלח אליה חיובי ; 0 אם הוא אפס ; -1 אם הוא שלילי	✓ 1. הפונקציה מחזירה את מחצית הערך שנשלח אליה, מעוגל כלפי מטה
✗ 5. יש לציין ערך מוחזר בעת שימוש במשפט return מפונקציה שמחזירה ערך	✓ 2. הפונקציה מחזירה את הערך שנשלח אליה (שים לב כי אופרטור ההעלאה העצמית מופיע לאחר שם המשתנה)
✗ 6. יש להחזיר ערך תוך שימוש במשפט return מפונקציה שמחזירה ערך	✗ 3. אין להחזיר ערך מפונקציה שאינה מחזירה ערך (void)

.12

ההצהרות הנכונות וההגדרות המתאימות להן :

1. ההצהרה מתאימה להגדרה 3
3. ההצהרה מתאימה להגדרה 1
4. ההצהרה מתאימה להגדרה 5
6. ההצהרה מתאימה להגדרה 2
7. ההצהרה מתאימה להגדרה 4

ההצהרות השגויות :

✗ 2. יש להוסיף סוגריים פותחים וסוגרים לאחר שם הפונקציה
✗ 5. יש להוסיף נקודה-פסיק לאחר הסוגריים

.13

1.

```
#include <stdio.h>

int GetDiv(int a, int b, int c);

void main()
{
    int num1, num2, num3, n;

    printf("Please enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);
```

```

    n = GetDiv(num1, num2, num3);
    printf("Result: %d \n", n);
}

int GetDiv(int a, int b, int c)
{
    int i, count = 0;

    for (i = a + 1 ; i < b ; i++)
        if (i % c == 0)
            count++;

    return count;
}

```

2.

```

#include <stdio.h>

int GetDigit(int x);

void main()
{
    int i, n = 0, num;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++) {
        scanf("%d", &num);
        n += GetDigit(num);
    }

    printf("%d digits are odd. \n", n);
}

int GetDigit(int x)
{
    int digit, count = 0;

    while (x > 0) {
        digit = x % 10;
        if (digit % 2 == 1)
            count++;
        x /= 10;
    }

    return count;
}

```

3.

```
#include <conio.h>

int IsCharLetter(char ch);

void main()
{
    char ch;

    do {
        ch = getch();
        if (IsCharLetter(ch) == 1)
            putchar(ch);
    } while (ch != ' ');
}

int IsCharLetter(char ch)
{
    if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')
        return 1;
    return 0;
}
```

פרק 9

מערכים

שאלות

מערכים חד-מימדיים

1. אחדות משורות ההגדרה שלפניך שגויות. סמן אותן וכתוב אותן כך שיהיו נכונות מבחינה תחבירית:

1. `int array[10];`
2. `char [40]chs;`
3. `float values[5.5];`
4. `int x, many_x[100], y, many_y[100];`
5. `int g = 20, many_g[g];`
6. `char letters['4'], digits;`
7. `float heights_of_student[1];`
8. `char characters[0];`
9. `char digits[90 / 9], letters[13 * 2];`
10. `int width, width[10];`

2. אחדים מקטעי התוכניות שלפניך שגויים. סמן אותם והסבר מהן הטעויות. עבור קטעי התוכניות הנכונים, ציין מהם ערכי איברי המערכים בסופם.

1.

```
int numbers[5];
numbers[0] = 0;
numbers[1] = 1;
numbers[2] = 2;
numbers[4] = 3;
numbers[3] = 4;
```

2.

```
char capital_a[6];
capital_a = 'A';
```

3.

```
int specials[3];
specials[0] = 1;
specials[specials[0]] = 3;
specials[specials[specials[0]] - 1] = 2;
```

4.

```
float speeds[7];
int i;
speeds[0] = 5.0;
for (i = 1 ; i < 7 ; i++)
    [speeds]i = d;
```

5.

```
int array1[5], array2[5], i;
for (i = 0 ; i < 5; i++)
    array1[i] = i * i;
for (i = 4 ; i > -1; i--)
    array2[i] = array1[4 - i];
```

6.

```
float boys[3], girls[3], d;
for (d = 0.0 ; d < 3.0 ; d++) {
    boys[d] = d + d - d / 2.0;
    girls[d] = boys[d] / 4;
}
```

7.

```
int ri[10], q = 10, m = 4;
do {
    ri[q - 1] = q - 1;
    q--;
    m += q + 2;
    ri(q) += q * m;
} while (m < q && q - 1 > 0);
```

8.

```
int q[5], r[5], i = 4, j;  
do {  
    q[i] = i;  
    r[i] = q[q[i]] * i;  
    i--;  
} while (i >= 0);
```

3. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש 10 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תכפיל את ערך כל איבר באינדקס שלו, ותפלוט את התוצאה.
2. תוכנית הקולטת מהמשתמש 10 מספרים ומאחסנת אותם במערך מתאים. התוכנית תקלוט מהמשתמש מספר נוסף ותפלוט את כל איברי המערך הקטנים ממנו.
3. תוכנית הקולטת מהמשתמש 20 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תפלוט את ערך האיבר הגבוה ביותר ואת ערך האיבר הנמוך ביותר.

4. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש 20 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תקלוט מהמשתמש מספר נוסף, ותפלוט כמה פעמים הוא מופיע במערך.
2. תוכנית הקולטת מהמשתמש 10 מספרים שלמים ומאחסנת אותם במערך מתאים. החל מהאיבר השני, התוכנית תוסיף לכל איבר את סכום האיברים הקודמים לו. לבסוף, תפלוט התוכנית את איברי המערך.
3. תוכנית הקולטת מהמשתמש 10 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תחליף את סדר ערכי האיברים שבמערך ותפלוט אותם.

5. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש 10 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תכפיל כל ערך איבר בערך האיבר שאחריו, ואת ערך האיבר האחרון בערך האיבר הראשון (לפני שהוכפל בערך האיבר השני). לבסוף, תפלוט התוכנית את ערכי איברי המערך.
2. תוכנית הקולטת מהמשתמש 20 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תקלוט שני מספרים המציינים שני אינדקסים, ותחליף בין ערכי שני האיברים המתאימים להם. קליטת שני המספרים וההחלפה תבוצע שוב ושוב, עד שיוזנו שני מספרים זהים. לבסוף, תפלוט התוכנית את ערכי איברי המערך.

3. תוכנית הקולטת מהמשתמש 15 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תקלוט מספר טבעי n ותזיז את כל איברי המערך n מקומות ימינה. לבסוף, תפלוט התוכנית את ערכי איברי המערך.

6. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש 10 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תעביר את כל האיברים שהאינדקס שלהם זוגי למערך אחר, ותפלוט את איבריו.

2. תוכנית הקולטת מהמשתמש 20 מספרים שלמים ומאחסנת אותם במערך מתאים. התוכנית תעביר את כל האיברים שהאינדקס שלהם זוגי לתחילתו של מערך אחר, ואת כל האיברים שהאינדקס שלהם אי-זוגי לסוף המערך האחר. לבסוף, תפלוט התוכנית את ערכי איברי המערך האחר.

3. תוכנית הקולטת מהמשתמש 10 מספרים שלמים למערך אחד ו-10 מספרים שלמים נוספים למערך שני. התוכנית תכפיל כל זוג איברים מבין שני המערכים שלהם אינדקס זהה, ותציב את התוצאה באיבריו של מערך שלישי, אף הוא בן 10 איברים. לבסוף, תפלוט התוכנית את ערכי איברי המערך השלישי.

7. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספרים ומאחסנת אותם בשני מערכים שונים בני 5 איברים, המייצגים וקטורים. התוכנית תבצע מכפלה וקטורית בין שני המערכים ותפלוט את התוצאה. כלומר, אם הווקטורים הם a ו- b , יחושב הביטוי:

$$a[0] * b[0] + a[1] * b[1] + a[2] * b[2] + a[3] * b[3] + a[4] * b[4]$$

2. תוכנית הקולטת מהמשתמש מספרים טבעיים ובודקת האם הם מתחלקים ב-3 וב-5 ללא שארית; המספרים המתחלקים ב-3 ללא שארית יוצבו באיברי המערך `threes`, והמספרים המתחלקים ב-5 ללא שארית יוצבו באיברי המערך `fives`. שני המערכים מכילים 5 איברים. לאחר שיוצבו ערכים מתאימים בכל איברי המערכים, הם ייפלטו.

3. תוכנית הקולטת מהמשתמש מספרים שלמים לשני מערכים שונים בני 10 איברים, ומאחסנת במערך שלישי רק את המספרים המופיעים בשני המערכים. לבסוף, ייפלטו איברי המערך השלישי.

8. כתוב את התוכניות הבאות :

- 100 מעבורות עוגנות ב-100 מזחים, הממוספרים מ-1 עד 100, בגדה המערבית של הנהר. ביום הראשון, כל המעבורות שמספר המזח שלהן מתחלק ב-1 ללא שארית עוברות לגדה שממול. ביום השני, כל המעבורות שמספר המזח שלהן מתחלק ב-2 ללא שארית עוברות לגדה שממול. ביום השלישי, כל המעבורות שמספר המזח שלהן מתחלק ב-3 ללא שארית עוברות לגדה שממול, וכן הלאה. מצא באילו מעבורות ניתן לחצות את הנהר מהגדה המזרחית לאחר היום ה-100. שים לב כי הספינות עוברות מגדה לגדה יותר מפעם אחד במהלך 100 הימים.
2. דרך עתיקת יומין למציאת מספרים ראשוניים בין 1 ל- n היא כזו: ראשית, רשום בשורה את כל המספרים הטבעיים בין 1 ל- n . עתה, מחק בקו את 1, משום שהוא אינו ראשוני. מצא את המספר הבא שאינו מחוק, 2, ומחק בקו את כל המספרים המתחלקים בו (פרט הוא עצמו). מצא את המספר הבא שאינו מחוק, 3, ומחק בקו את כל המספרים המתחלקים בו (פרט הוא עצמו). מצא את המספר הבא שאינו מחוק, 5, ומחק בקו את כל המספרים המתחלקים בו (פרט הוא עצמו). המשך למחוק מספרים באופן זה, עד למציאת המספר המקסימלי שאינו מחוק בטווח בין 1 ל- n . עתה, המספרים שהוקפו הם המספרים הראשוניים בין 1 ל- n . כתוב תוכנית המסתמכת על טכניקה זו ופולטת את המספרים הראשוניים בין 1 ל-100.
3. תוכנית הקולטת מהמשתמש 20 אותיות קטנות, ופולטת כמה פעמים נקלטה כל אות.

מערכים דו-מימדיים

9. מוגדר מערך דו-מימדי שלם בשם `nums` בעל חמש שורות וארבע עמודות. ערכי איבריו מתאימים לערכים שבטבלה הבאה:

1	4	4	0
5	6	9	8
3	7	4	1
8	6	5	6
4	0	6	9

לפניך מספר קטעי תוכניות הפונים אל איברי מערך זה. ציין מהו הפלט המתקבל מכל אחד מהם. התייחס אל המשתנים i ו- j כאל משתנים שלמים.

1.

```
for (i = 0 ; i < 5 ; i++)  
    printf("-%d- ", nums[i][1]);
```

2.

```
for (j = 0 ; j < 5 ; j++)  
    printf("%d ", nums[j][0] + nums[1][j/2]);
```

3.

```
for (i = 0 ; i < 3 ; i++)  
    for (j = 2 ; j < 4 ; j++)  
        printf("%d", nums[j][i]);
```

4.

```
for (i = 0 ; i < 4 ; i++) {  
    printf("%d ", nums[3][i]);  
    if (nums[2][i + 1] == 7)  
        i--;  
    else  
        i = 3;  
}
```

5.

```
for (i = 0 ; i < 2 ; i++) {  
    for (j = 2 ; j < 4 ; j++)  
        printf("%d", nums[j][i]);  
    printf("%d ", nums[j][j - i - 1]);  
}
```

6.

```
for (i = 1 ; i < 4 ; i++)  
    for (j = 3 ; j > 1 ; j--)  
        printf("%d ", nums[nums[0][i]][j]);
```

7.

```
for (i = 3 ; i < 5 ; i++)  
    for (j = 1 ; j < 4 ; j++)  
        printf("%d", nums[i % j][nums[0][j]/2]);
```

8.

```
j = 0;  
for (i = 0 ; i < 4 ; i++) {  
    j += i;  
    printf("'%d'", nums[j % 5][i]);  
    if (nums[j % 5][i] < j * 2)  
        j = 0;  
}
```

10. כתוב את התוכניות הבאות :

1. מכל שלושה מספרים a , b ו- c - ניתן ליצור ריבוע קסם בן 3 שורות ו-3 עמודות, שאיבריו הם אלו :

$a + b$	$a - (b + c)$	$a + c$
$a - (b - c)$	a	$a + (b - c)$
$a - c$	$a + (b + c)$	$a - b$

כתוב תוכנית הקולטת מהמשתמש את הערכים a , b ו- c (כולם ערכים שלמים), ומציגה את ריבוע הקסם המתאים להם.

2. על פי התוכנית הקודמת, כתוב תוכנית הקולטת מהמשתמש את הערכים a , b ו- c ויוצרת ריבוע קסם המורכב מחזקות של 2, כשסכום מעריכי החזקה בכל טור, שורה ואלכסון שווה. למשל, ריבוע הקסם הבא מקיים את התנאי :

2^5	2^0	2^4
2^2	2^3	2^4
2^2	2^6	2^1

התוכנית תפלוט את ערך החזקות ; למשל, עבור ריבוע הקסם דלעיל, תפלוט התוכנית את ריבוע הקסם הבא :

32	1	16
4	8	16
4	64	2

שים לב שהערכים a , b ו- c חייבים להיות אי-שליליים ושלמים, ולקיים את התנאי $a \geq b + c$. כמו-כן, מכיון שמדובר בחזקות של 2, יש להיזהר שלא תתרחש גלישה מטווח הערכים של המשתנה השלם (מומלץ להגדיר את איברי המערך כמספרים מסוג long, כדי להגדיל את טווח הערכים האפשריים).

3. תוכנית הקולטת מהמשתמש מספרים ומאחסנת אותם במערך בן 6 שורות ו-4 עמודות. התוכנית תיחשב את סכום האיברים החיצוניים ביותר (כלומר, את האיברים שמיקומם הוא היקף המערך).

11. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מספר ובונה מערך דו-מימדי בן 10 שורות ו-10 עמודות, המכיל את לוח הכפל החל ממספר זה. למשל, עבור המספר 15, ייבנה המערך הבא :

225	240	255	270	285	300	315	330	345	360
240	256	272	288	304	320	336	352	368	384
:	:	:	:	:	:	:	:	:	:
360	384	408	432	456	480	504	528	552	576

התוכנית תפלוט את איברי המערך באופן קריא וברור.

2. תוכנית הקולטת מהמשתמש מספרים שלמים ומאחסנת אותם בשני מערכים דו-מימדיים בני 5 שורות ו-5 עמודות. התוכנית תקלוט מספר שלם מהמשתמש, בין 0 ל-4, המייצג את אחת משורות המערכים, ותחליף את איברי שורה זו בין שני המערכים. לבסוף, ייפלטו איברי השורות שהוחלפו.

3. מפעל מנהל את משכורות עובדיו על ידי טבלה, בה כל שורה מייצגת עובד אחר, וכל עמודה – יום אחר בשבוע. בכל תא בטבלה רשום מספר השעות שעבד עובד כלשהו ביום מסוים. במפעל 10 עובדים, העובדים חמישה ימים בשבוע. כתוב תוכנית הקולטת את מספר השעות שעבד כל עובד בכל יום ומאחסנת במערך הדו-מימדי נתונים אלה. לאחר מכן תאחסן התוכנית במערך נוסף את מספר השעות שעבד כל עובד במשך שבוע עבודתו. לבסוף תפלוט התוכנית את המשכורת שיש לשלם לכל עובד בסוף השבוע, אם עבור כל שעה שעבד הוא מקבל 122 ש"ח.

12. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מספרים למערך דו-מימדי בן ארבע שורות וארבע עמודות, ומוצאת האם המערך הוא ריבוע קסם. למשל, הנה ריבוע קסם בן ארבע שורות וארבע עמודות :

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

2. תוכנית הקולטת מהמשתמש מספרים למערך דו-מימדי בן שלוש שורות ושלוש עמודות, ופולטת מלל מתאים אם **מכפלת** האיברים בכל שורה, בכל טור ובאלכסון הראשי והמשני שווה. למשל, המערך הדו-מימדי הבא מקיים את התנאי:

18	1	12
4	6	9
3	36	2

3. תוכנית הקולטת מספרים למערך דו-מימדי בן שלוש שורות ושלוש עמודות, ופולטת מלל מתאים אם המערך הוא ריבוע קסם שכל איבריו ראשוניים. למשל, המערך הדו-מימדי הבא מקיים את התנאי:

83	29	101
89	71	53
41	113	59

13. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מספרים למערך דו-מימדי בן שלוש שורות ושלוש עמודות, המייצג מטריצה, ולמערך חד-מימדי בן שלושה איברים, המייצג וקטור. התוכנית תבצע מכפלה וקטורית בין השניים ותפלוט את איברי וקטור התוצאה. כלומר, אם המטריצה היא M , הווקטור הוא V והווקטור התוצאה הוא S , אזי:

$$S[0] = M[0][0] * V[0] + M[0][1] * V[1] + M[0][2] * V[2]$$

$$S[1] = M[1][0] * V[0] + M[1][1] * V[1] + M[1][2] * V[2]$$

$$S[2] = M[2][0] * V[0] + M[2][1] * V[1] + M[2][2] * V[2]$$

2. תוכנית הקולטת מספר טבעי אי-זוגי n קטן מ-10, וממלאת בכוכביות את איבריו של מערך דו-מימדי מתאים, כך שצורתן תיצור משולש שווה שוקיים שבסיסו בנוי מ- n כוכביות. שאר איברי המערך יכילו תו רווח. לאחר מכן, התוכנית תפלוט את איברי המערך, כך שיוצג המשולש המתאים. למשל, אם יוזן המספר 5, ייפלט המשולש הבא:

```
*
***
*****
```

3. תוכנית הקולטת מהמשתמש את מראהו של לוח איקס-עיגול בן 3 שורות ו-3 עמודות, ומוצאת מי משני השחקנים, אם בכלל, ניצח. התוכנית תקלוט בכל פעם שורה אחרת בלוח האי-קס-העיגול (מומלץ להשתמש בפונקציה getch), כאשר שחקן אחד הוא 'X' ושחקן אחר הוא 'O'. למשל, עבור הקלט הבא:

```
X O O
X O X
X X O
```

יוצג הפלט הבא:

Player X wins!

אתחול מערכים

אתחול מערך חד-מימדי

14. אחדות משורות ההגדרה שלפניך שגויות. סמן אותן והסבר מהן הטעויות:

1. `int primes[6] = { 2, 3, 5, 7, 11, 13 };`
2. `int odds[7] = 1, 3, 5, 7, 9, 11, 13;`
3. `char vowels[5] = { 'a' 'e' 'i' 'o' 'u' };`
4. `float ages[5] = { 17.5, , 33.4, , 13.4 };`
5. `char marks[3] = { '*', '^', '$' };`
6. `float fractions[5] = { 0.3, 0.7, 0.1 };`
7. `int fib[5] = { 0, 1, 1, 2, 3, 5, 8, 13 };`
8. `int pal[] = { 0, 14341, 5660, 212, 78 };`
9. `char letters[] = { 0 };`
10. `char letters[26] = 0;`
11. `char letters[26] = { 0 };`
12. `float q, p, ri[4] = { 1.0, 2.0 }, qp;`

אתחול מערך דו-מימדי

15. אחדות משורות ההגדרה שלפניך שגויות. סמן אותן והסבר מהן הטעויות. עבור שורות ההגדרה הנכונות, ציין את ערכי איברי המערכים לאחר ביצוען.

1.

```
int mults[3][4] = { 2, 4, 6, 8,
                   3, 6, 9, 12,
                   4, 8, 12, 16 };
```

2.

```
int perfect[4][3] = { 1, 2, 3, 4,
                     5, 6, 7, 8,
                     9, 10, 11, 12 };
```

3.

```
int values[5][2] = { 0, 1, 1, 2, 4, 7,
                    13, 24, 44, 81 };
```

4.

```
int threes[4][2] = { 0, 3, 6, 9, 12,
                    15, 18, 21, 24 };
```

5.

```
float numbers[1][5] = { 1.0, 2.0, 3.0 };
```

6.

```
int breaks[2][2] = { 0 };
```

7.

```
char stuff[2, 4] = { '#', '^', '\\', '\\',
                    '*', '&', '%', '$' };
```

8.

```
char letters[4][1] = 'a', 'b', 'c', 'd';
```

9.

```
int ri[3][5] = { { 1, 1, 1, 1, 1 },
                 { 2, 2, 2, 2, 2 },
                 { 3, 3, 3, 3, 3 } };
```

10.

```
int ri[3][5] = { { 1, 1, 1, 1, 1 }
                 { 2, 2, 2, 2, 2 }
                 { 3, 3, 3, 3, 3 } };
```

11.

```
int ri[3][5] = { { { 1, 1, 1, 1, 1 },  
                 { 2, 2, 2, 2, 2 },  
                 { 3, 3, 3, 3, 3 } } };
```

12.

```
float ri[3][5] = { 1, 1, 1.5, 1, 1 },  
                 { 2, 2, 2.5, 2, 2 },  
                 { 3, 3, 3.5, 3, 3 };
```

13.

```
int reducibles[3][4] = { { 4, 6, 8 } ,  
                        { 9, 10, 12 },  
                        { 14, 15, 16 },  
                        { 18, 20, 21 } };
```

14.

```
int tic[3][3] = { { 0, 1 },  
                 { 1, 1 },  
                 { 1 } };
```

15.

```
int tic[3][3] = { 0, 1,  
                 1, 1,  
                 1 };
```

16.

```
int factorials[][4] = { 0, 1, 2, 6,  
                       24, 120, 720 };
```

17.

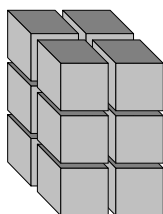
```
float sqrts[2][] = { 1.00, 1.41, 1.73,  
                   2.00, 2.23, 2.45,  
                   2.64, 2.82, 3.00 };
```

18.

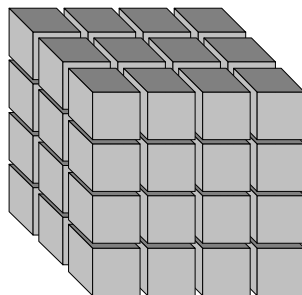
```
float sqrs[][] = { 1.00, 2.25, 4.00, 6.25,  
                  9.00, 12.25, 16, 20.25 };
```

מערכים רב-מימדיים

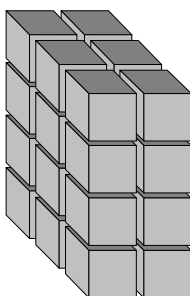
16. לפניך איורים המייצגים מערכים תלת-מימדיים. רשום תחת כל איור שורת הגדרה המתאימה למערך המוצג בו.



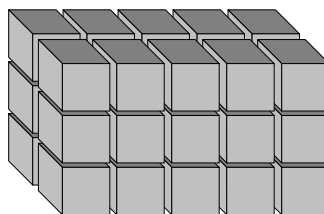
2



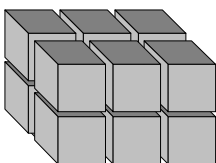
1



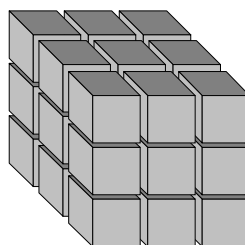
4



3

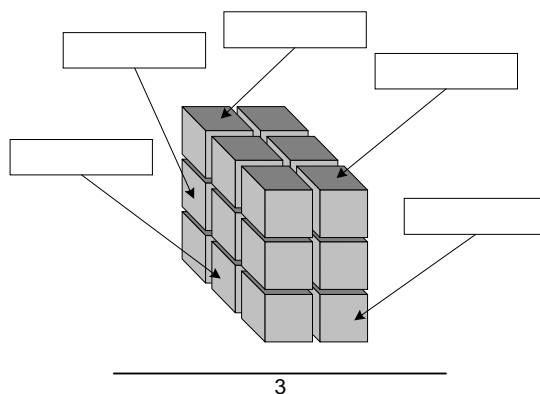
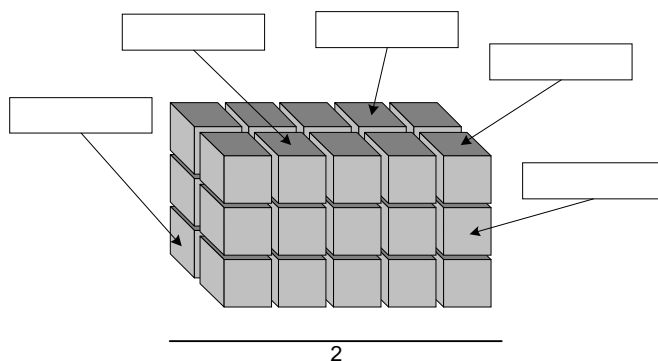
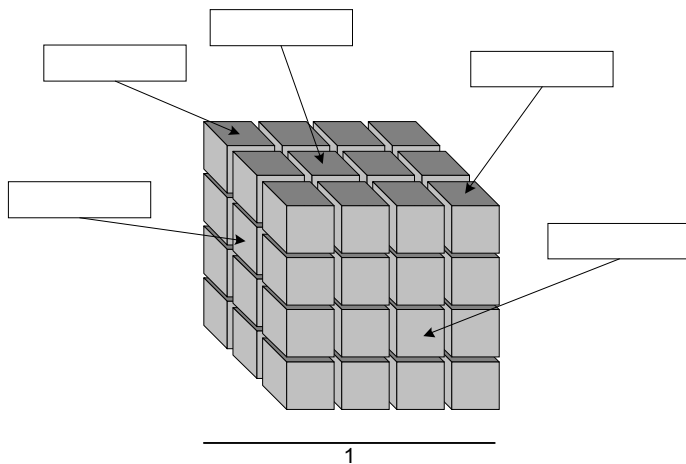


6



5

17. לפניך מספר מערכים תלת-מימדיים ואיברים נבחרים בהם. רשום תחת כל חץ את מיקומו של האיבר אליו הוא מצביע.



אתחול מערך רב-מימדי

18. בטבלאות שלפניך רשומים איברי מערכים רב-מימדיים וערכיהם. התאם לכל טבלה שורת הגדרה מתאימה וקצרה ככל הניתן, שלאחריה יכילו איברי המערך את הערכים הרשומים בה.

איבר	ערך	איבר	ערך
[1][0][0]	11	[0][0][0]	2
[1][0][1]	13	[0][0][1]	3
[1][1][0]	17	[0][1][0]	5
[1][1][1]	19	[0][1][1]	7

איבר	ערך	איבר	ערך
[1][1][0]	20	[0][0][0]	0
[1][1][1]	37	[0][0][1]	1
[2][0][0]	0	[0][1][0]	2
[2][0][1]	0	[0][1][1]	3
[2][1][0]	0	[1][0][0]	6
[2][1][1]	0	[1][0][1]	11

איבר	ערך	איבר	ערך
[1][1][0]	0	[0][0][0]	1
[1][1][1]	0	[0][0][1]	2
[1][1][2]	0	[0][0][2]	3
[2][0][0]	7	[0][1][0]	0
[2][0][1]	8	[0][1][1]	0
[2][0][2]	9	[0][1][2]	0
[2][1][0]	0	[1][0][0]	4
[2][1][1]	0	[1][0][1]	5
[2][1][2]	0	[1][0][2]	6

איבר	ערך	איבר	ערך	איבר	ערך
[1][0][2][0]	0	[0][1][1][0]	0	[0][0][0][0]	100
[1][0][2][1]	0	[0][1][1][1]	0	[0][0][0][1]	200
[1][1][0][0]	0	[0][1][2][0]	0	[0][0][1][0]	300
[1][1][0][1]	0	[0][1][2][1]	0	[0][0][1][1]	200
[1][1][1][0]	0	[1][0][0][0]	0	[0][0][2][0]	100
[1][1][1][1]	0	[1][0][0][1]	0	[0][0][2][1]	0
[1][1][2][0]	0	[1][0][1][0]	0	[0][1][0][0]	0
[1][1][2][1]	0	[1][0][1][1]	0	[0][1][0][1]	0

איבר	ערך	איבר	ערך	איבר	ערך
[1][0][1][1]	0	[0][1][0][2]	0	[0][0][0][0]	123
[1][0][1][2]	0	[0][1][1][0]	0	[0][0][0][1]	0
[1][1][0][0]	0	[0][1][1][1]	0	[0][0][0][2]	0
[1][1][0][1]	0	[0][1][1][2]	0	[0][0][1][0]	0
[1][1][0][2]	0	[1][0][0][0]	789	[0][0][1][1]	0
[1][1][1][0]	0	[1][0][0][1]	0	[0][0][1][2]	0
[1][1][1][1]	0	[1][0][0][2]	0	[0][1][0][0]	456
[1][1][1][2]	0	[1][0][1][0]	0	[0][1][0][1]	0

שליחת מערכים לפונקציות

שליחת מערך חד-מימדי

19. הקף בעיגול את המשפטים הנכונים:

1. בעת הגדרת מערך חד-מימדי כמשתנה פורמאלי חובה לציין את מספר איבריו.
2. אל פונקציה המקבלת מערך חד-מימדי כפרמטר ניתן לשלוח כל מערך חד-מימדי מאותו סוג.
3. פונקציה המקבלת מערך חד-מימדי כפרמטר אינה יכולה לדעת את מספר איבריו בלי שישלח אליה פרמטר נוסף המציין זאת.
4. פונקציה המקבלת מערך כפרמטר יכולה לשנות את ערכי איבריו.

20. כתוב את הפונקציות הבאות :

1. פונקציה המקבלת כפרמטרים מערך חד-מימדי שלם ומספר n הקובע את מספר איבריו, ופולטת את כל איבריו.
2. פונקציה המקבלת כפרמטרים מערך חד-מימדי שלם ומספר n הקובע את מספר איבריו, קולטת מהמשתמש n מספרים ומאחסנת אותם באיברי המערך. הפונקציה תחזיר את סכום המספרים שהוזנו.
3. פונקציה המקבלת כפרמטרים שני מערכים חד-מימדיים שלמים בגודל זהה ומספר n הקובע את מספר איבריהם, ומאחסנת במערך החד-מימדי השני את ריבועי ערכי האיברים שבמערך החד-מימדי הראשון.

שליחת מערך דו-מימדי

21. הקף בעיגול את המשפטים הנכונים :

1. לפונקציה המקבלת מערך דו-מימדי כפרמטר ניתן לשלוח כל מערך חד-מימדי מאותו סוג.
2. לפונקציה המקבלת מערך דו-מימדי כפרמטר ניתן לשלוח כל מערך דו-מימדי מאותו סוג.
3. אין חשיבות למספר השורות (גודל המימד הראשון) של מערך דו-מימדי המוגדר כפרמטר של פונקציה.
4. ניתן לשלוח אל פונקציה מערך דו-מימדי שמספר שורותיו ועמודותיו אינו קבוע.

22. כתוב את הפונקציות הבאות :

1. פונקציה המקבלת כפרמטרים מערך דו-מימדי שלם בעל 4 עמודות ומספר n הקובע את מספר שורותיו, קולטת מהמשתמש ערכים ומאחסנת אותם במערך.
2. פונקציה המקבלת כפרמטרים מערך דו-מימדי שלם בעל 5 עמודות ומספר n הקובע את מספר שורותיו, ומחזירה את ערך האיבר הגדול ביותר במערך.
3. פונקציה המקבלת כפרמטרים מערך דו-מימדי שלם בעל 4 עמודות, מספר n הקובע את מספר שורותיו, ומערך חד-מימדי שלם בעל 4 איברים. התוכנית תציב בכל אחד מאיברי המערך החד-מימדי את סכום האיברים שבעמודות המערך הדו-מימדי.

תשובות

.1

	.6 ✓		.1 ✓
	.7 ✓	char chs[40];	.2 ✗
char characters[1];	.8 ✗	float values[6];	.3 ✗
	.9 ✓		.4 ✓
int width, width2[10];	.10 ✗	int g = 20, many_g[20];	.5 ✗

.2

array1: 0, 1, 4, 9, 16	.5 ✓	0, 1, 2, 4, 3	.1 ✓
array2: 16, 9, 4, 1, 0			
אין לפנות אל איבר במערך על ידי ערך עשירי	.6 ✗	אין להציב ערך במערך מבלי לפנות לאיבר מסוים בו	.2 ✗
יש לציין את אינדקס האיבר בסוגריים מרובעים	.7 ✗	1, 3, 2	.3 ✓
q: 0, 1, 2, 3, 4	.8 ✓	בתוך הסוגריים המרובעים יש לרשום את אינדקס האיבר אליו רוצים לפנות	.4 ✗
r: 0, 1, 4, 9, 16			

.3

1.

```
#include <stdio.h>
void main()
{
    int array[10], i;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%d", &array[i]);

    for (i = 0 ; i < 10 ; i++) {
        array[i] *= i;
        printf("%d ", array[i]);
    }
}
```

2.

```
#include <stdio.h>
void main()
{
    float array[10], num;
    int i;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%f", &array[i]);

    printf("Please enter a number: ");
    scanf("%f", &num);

    for (i = 0 ; i < 10 ; i++)
        if (array[i] < num)
            printf("%f ", array[i]);
}
```

3.

```
#include <stdio.h>
void main()
{
    int array[20], highest = 0, lowest = 0, i;

    printf("Please enter 20 numbers: ");
    for (i = 0 ; i < 20 ; i++)
        scanf("%d", &array[i]);

    for (i = 1 ; i < 20 ; i++) {
        if (array[i] > array[highest])
            highest = i;
        if (array[i] < array[lowest])
            lowest = i;
    }

    printf("Highest: %d \n", array[highest]);
    printf("Lowest: %d \n", array[lowest]);
}
```

1.

```

#include <stdio.h>
void main()
{
    int array[20], i, num, count = 0;

    printf("Please enter 20 numbers: ");
    for (i = 0 ; i < 20 ; i++)
        scanf("%d", &array[i]);

    printf("Please enter a number to find: ");
    scanf("%d", &num);

    for (i = 0 ; i < 20 ; i++)
        if (array[i] == num)
            count++;

    printf("%d appears %d times in the array. \n", num, count);
}

```

2.

```

#include <stdio.h>
void main()
{
    int array[10], i;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%d", &array[i]);

    printf("%d ", array[0]);
    for (i = 1 ; i < 10 ; i++) {
        array[i] += array[i - 1];
        printf("%d ", array[i]);
    }
}

```

3.

```

#include <stdio.h>
void main()
{
    int array[10], n, temp;

    printf("Please enter ten numbers: ");

```

```

for (n = 0 ; n < 10 ; n++)
    scanf("%d", &array[n]);

for (n = 0 ; n < 5 ; n++) {
    temp = array[n];
    array[n] = array[9 - n];
    array[9 - n] = temp;
}

printf("Numbers in reversed order: ");
for (n = 0 ; n < 10 ; n++)
    printf("%d ", array[n]);
}

```

.5

1.

```

#include <stdio.h>
void main()
{
    int array[10], temp, i;

    printf("Please enter ten numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%d", &array[i]);

    temp = array[0];
    for (i = 0 ; i < 9 ; i++)
        array[i] *= array[i + 1];
    array[9] *= temp;

    for (i = 0 ; i < 10 ; i++)
        printf("%d ", array[i]);
}

```

2.

```

#include <stdio.h>
void main()
{
    int array[20], temp, index1, index2, i;

    printf("Please enter 20 numbers: ");
    for (i = 0 ; i < 20 ; i++)
        scanf("%d", &array[i]);
}

```

```

printf("Please enter pairs of indexes: \n");
scanf("%d %d", &index1, &index2);

while (index1 != index2) {
    temp = array[index1];
    array[index1] = array[index2];
    array[index2] = temp;

    scanf("%d %d", &index1, &index2);
}

for (i = 0 ; i < 20 ; i++)
    printf("%d ", array[i]);
}

```

3.

```

#include <stdio.h>
void main()
{
    int array[15], temp, i, n;

    printf("Please enter 15 numbers: ");
    for (i = 0 ; i < 15 ; i++)
        scanf("%d", &array[i]);

    printf("How many shifts to the right? ");
    scanf("%d", &n);

    for ( ; n > 0 ; n--) {
        temp = array[14];
        for (i = 14 ; i > 0 ; i--)
            array[i] = array[i - 1];
        array[0] = temp;
    }

    for (i = 0 ; i < 15 ; i++)
        printf("%d ", array[i]);
}

```

1.

```
#include <stdio.h>
void main()
{
    int array1[10], array2[5], i;

    printf("Please enter 10 numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%d", &array1[i]);

    for (i = 0 ; i < 5 ; i++) {
        array2[i] = array1[i * 2];
        printf("%d ", array2[i]);
    }
}
```

2.

```
#include <stdio.h>
void main()
{
    int array1[20], array2[20], i;

    printf("Please enter 20 numbers: ");
    for (i = 0 ; i < 20 ; i++)
        scanf("%d", &array1[i]);

    for (i = 0 ; i < 10 ; i++) {
        array2[i] = array1[i * 2];
        array2[i + 10] = array1[i * 2 + 1];
    }

    for (i = 0 ; i < 20 ; i++)
        printf("%d ", array2[i]);
}
```

3.

```
#include <stdio.h>
void main()
{
    int array1[10], array2[10], array3[10], i;

    printf("Please enter 10 numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%d", &array1[i]);
}
```

```

printf("Please enter 10 more numbers: ");
for (i = 0 ; i < 10 ; i++)
    scanf("%d", &array2[i]);

for (i = 0 ; i < 10 ; i++) {
    array3[i] = array1[i] * array2[i];
    printf("%d ", array3[i]);
}
}

```

.7

1.

```

#include <stdio.h>
void main()
{
    int a[5], b[5], i, result = 0;

    printf("Please enters five numbers: ");
    for (i = 0 ; i < 5 ; i++)
        scanf("%d", &a[i]);
    printf("Please enters five additional numbers: ");
    for (i = 0 ; i < 5 ; i++) {
        scanf("%d", &b[i]);
        result += a[i] * b[i];
    }

    printf("Vector multiplication result: %d \n", result);
}

```

2.

```

#include <stdio.h>
void main()
{
    int threes[5], fives[5], threes_count = 0, fives_count = 0,
        num, i;

    printf("Please enters some numbers: \n");
    do {
        scanf("%d", &num);
        if (num % 3 == 0 && threes_count < 5) {
            threes[threes_count] = num;
            threes_count++;
        }
    }
}

```

```

        if (num % 5 == 0 && fives_count < 5) {
            fives[fives_count] = num;
            fives_count++;
        }
    } while (threes_count < 5 || fives_count < 5);

    printf("Numbers that divide by 3 without remainder: ");
    for (i = 0 ; i < 5 ; i++)
        printf("%d ", threes[i]);
    printf("\nNumbers that divide by 5 without remainder: ");
    for (i = 0 ; i < 5 ; i++)
        printf("%d ", fives[i]);
}

```

3.

```

#include <stdio.h>
void main()
{
    int array1[10], array2[10], array3[10], i, j, count = 0,
        found;

    printf("Please enters ten numbers: ");
    for (i = 0 ; i < 10 ; i++)
        scanf("%d", &array1[i]);
    printf("Please enters ten additional numbers: ");
    for (i = 0 ; i < 10 ; i++) {
        scanf("%d", &array2[i]);

        /* see if array2[i] appears somewhere in array3 */
        j = 0;
        while (j < count && array3[j] != array2[i])
            j++;

        found = 0;
        if (j == count) /* then array2[i] is not in array3 */
            for (j = 0 ; j < 10 && found == 0; j++)
                if (array1[j] == array2[i]) {
                    array3[count] = array2[i];
                    count++;
                    found = 1;
                }
    }

    printf("Numbers that appear in both arrays: ");
    for (j = 0 ; j < count ; j++)
        printf("%d ", array3[j]);
}

```

1.

```

#include <stdio.h>
void main()
{
    int array[100], i, n;

    /* 1 <==> western shore
       * -1 <==> eastern shore
       */
    for (i = 0 ; i < 100 ; i++)
        array[i] = 1;

    for (i = 1 ; i <= 100 ; i++)
        for (n = i ; n <= 100 ; n += i)
            array[n - 1] *= -1;

    printf("Available ships after 100 days: ");
    for (i = 0 ; i < 100 ; i++)
        if (array[i] == -1)
            printf("%d ", i + 1);
}

```

2.

```

#include <stdio.h>
void main()
{
    int array[100], i, m = 2;
    /* 0 <==> erased numbers */
    for (i = 0 ; i < 100 ; i++)
        array[i] = 1;

    array[0] = 0;
    do {
        for (i = m - 1 + m ; i < 100 ; i += m)
            array[i] = 0;
        do
            m++;
        while (m < 100 && array[m - 1] == 0);
    } while (m < 100);

    printf("The prime numbers from 1 to 100 are: \n");
    for (i = 0 ; i < 100 ; i++)
        if (array[i] == 1)
            printf("%d ", i + 1);
}

```

3.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char array[26], ch, i;

    for (i = 0 ; i < 26; i++)
        array[i] = 0;

    printf("Please enter 20 small letters: ");
    for (i = 0 ; i < 20 ; i++) {
        ch = getch();
        putchar(ch);
        array[ch - 'a']++;
    }

    printf("The letters you've entered are: \n");
    for (i = 0 ; i < 26 ; i++)
        if (array[i] != 0)
            printf("%c -- %d times \n", 'a' + i, array[i]);
}
```

.9

389 766	.5	-4- -6- -7- -6- -0-	.1
9 6 9 6 0 4	.6	6 10 9 14 13	.2
491445	.7	387645	.3
*1*6*5*6	.8	8 8 8 8 ... (פלט אינסופי של 8)	.4

.10

1.

```
#include <stdio.h>
void main()
{
    int sq[3][3], row, col, a, b, c;

    printf("Please enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    sq[0][0] = a + b;
    sq[0][1] = a - (b + c);
}
```

```

sq[0][2] = a + c;
sq[1][0] = a - (b - c);
sq[1][1] = a;
sq[1][2] = a + (b - c);
sq[2][0] = a - c;
sq[2][1] = a + (b + c);
sq[2][2] = a - b;

for (row = 0 ; row < 3 ; row++) {
    for (col = 0 ; col < 3 ; col++)
        printf("%d \t", sq[row][col]);
    printf("\n");
}
}

```

2.

```

#include <stdio.h>
void main()
{
    int i, row, col, a, b, c;
    long sq[3][3], mult;

    printf("Please enter three numbers: \n");
    scanf("%d %d %d", &a, &b, &c);

    sq[0][0] = a + b;
    sq[0][1] = a - (b + c);
    sq[0][2] = a + c;
    sq[1][0] = a - (b - c);
    sq[1][1] = a;
    sq[1][2] = a + (b - c);
    sq[2][0] = a - c;
    sq[2][1] = a + (b + c);
    sq[2][2] = a - b;

    for (row = 0 ; row < 3 ; row++)
        for (col = 0 ; col < 3 ; col++) {
            mult = 1;
            for (i = sq[row][col] ; i > 0 ; i--)
                mult *= 2;
            sq[row][col] = mult;
        }
}

```

```

for (row = 0 ; row < 3 ; row++) {
    for (col = 0 ; col < 3 ; col++)
        printf("%ld \t", sq[row][col]);
    printf("\n");
}
}

```

3.

```

#include <stdio.h>
void main()
{
    int array[6][4], sum = 0, row, col;

    printf("Please enter 24 numbers: \n");
    for (row = 0 ; row < 6 ; row++)
        for (col = 0 ; col < 4 ; col++)
            scanf("%d", &array[row][col]);

    for (col = 0 ; col < 4 ; col++)
        sum += array[0][col] + array[5][col];

    for (row = 1 ; row < 5 ; row++)
        sum += array[row][0] + array[row][3];

    printf("Perimeter sum: %d \n", sum);
}

```

.11

1.

```

#include <stdio.h>
void main()
{
    int array[10][10], row, col, num;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (row = 0 ; row < 10 ; row++) {
        for (col = 0 ; col < 10 ; col++) {
            array[row][col] = (num + row) * (num + col);
            printf("%d \t", array[row][col]);
        }
        printf("\n");
    }
}

```

2.

```
#include <stdio.h>
void main()
{
    int array1[5][5], array2[5][5], row, col,
        line, temp;

    printf("Please enter 25 numbers: \n");
    for (row = 0 ; row < 5 ; row++)
        for (col = 0 ; col < 5 ; col++)
            scanf("%d", &array1[row][col]);

    printf("Please enter 25 additional numbers: \n");
    for (row = 0 ; row < 5 ; row++)
        for (col = 0 ; col < 5 ; col++)
            scanf("%d", &array2[row][col]);

    printf("Please select a line to exchange: ");
    scanf("%d", &line);

    for (col = 0 ; col < 5 ; col++) {
        temp = array1[line][col];
        array1[line][col] = array2[line][col];
        array2[line][col] = temp;
    }

    printf("Replaced values: \n");
    for (col = 0 ; col < 5 ; col++)
        printf("%d ", array1[line][col]);
    printf("\n");
    for (col = 0 ; col < 5 ; col++)
        printf("%d ", array2[line][col]);
}
```

3.

```
#include <stdio.h>
void main()
{
    int array[10][5], hours[10], row, col;

    printf("Please enter work hours for the employees: \n");
    for (row = 0 ; row < 10 ; row++) {
        hours[row] = 0;
        for (col = 0 ; col < 5 ; col++) {
            scanf("%d", &array[row][col]);
            hours[row] += array[row][col];
        }
    }
}
```

```

    }
}

for (row = 0 ; row < 10 ; row++)
    printf("The salary of employee #d: %d NIS \n", row + 1,
        hours[row] * 122);
}

```

.12

1.

```

#include <stdio.h>
void main()
{
    int sq[4][4], row, col, is_magic = 1, sum;

    printf("Please enter 16 numbers: \n");

    for (row = 0 ; row < 4 ; row++)
        for (col = 0 ; col < 4 ; col++)
            scanf("%d", &sq[row][col]);

    sum = sq[0][0] + sq[1][1] + sq[2][2] + sq[3][3];
    if (sq[0][3] + sq[1][2] + sq[2][1] + sq[3][0] != sum)
        is_magic = 0;

    if (is_magic == 1)
        for (row = 0 ; row < 4 ; row++)
            if (sq[row][0] + sq[row][1] + sq[row][2] + sq[row][3]
                != sum || sq[0][row] + sq[1][row] + sq[2][row] +
                sq[3][row] != sum)
                is_magic = 0;

    if (is_magic == 1)
        printf("This is a magic square! \n");
    else
        printf("This isn't a magic square! \n");
}

```

2.

```

#include <stdio.h>
void main()
{
    int sq[3][3], row, col, is_magic = 1, mult;

```

```

printf("Please enter nine numbers: \n");

for (row = 0 ; row < 3 ; row++)
    for (col = 0 ; col < 3 ; col++)
        scanf("%d", &sq[row][col]);

mult = sq[0][0] * sq[1][1] * sq[2][2];
if (sq[0][2] * sq[1][1] * sq[2][0] != mult)
    is_magic = 0;

if (is_magic == 1)
    for (row = 0 ; row < 3 ; row++)
        if (sq[row][0] * sq[row][1] * sq[row][2] != mult ||
            sq[0][row] * sq[1][row] * sq[2][row] != mult)
            is_magic = 0;

if (is_magic == 1)
    printf("This is a multiplication magic square! \n");
else
    printf("This isn't a multiplication magic square! \n");
}

```

3.

```

#include <stdio.h>
void main()
{
    int sq[3][3], row, col, is_magic = 1, sum, num, n;

    printf("Please enter nine numbers: \n");

    for (row = 0 ; row < 3 ; row++)
        for (col = 0 ; col < 3 ; col++) {
            scanf("%d", &sq[row][col]);
            num = sq[row][col];
            if (num < 0)
                num = -num;
            if (num < 2)
                is_magic = 0;
        }
    else
        for (n = 2 ; n <= num / 2 && is_magic == 1 ; n++)
            if (num % n == 0)
                is_magic = 0;
}

```

```

sum = sq[0][0] + sq[1][1] + sq[2][2];
if (sq[0][2] + sq[1][1] + sq[2][0] != sum)
    is_magic = 0;

if (is_magic == 1)
    for (row = 0 ; row < 3 ; row++)
        if (sq[row][0] + sq[row][1] + sq[row][2] != sum ||
            sq[0][row] + sq[1][row] + sq[2][row] != sum)
            is_magic = 0;

if (is_magic == 1)
    printf("This is a prime magic square! \n");
else
    printf("This isn't a prime magic square! \n");
}

```

.13

1.

```

#include <stdio.h>
void main()
{
    int matrix[3][3], vector[3], result[3];
    int row, col;

    printf("Please enter matrix values: \n");
    for (row = 0 ; row < 3 ; row++)
        for (col = 0 ; col < 3 ; col++)
            scanf("%d", &matrix[row][col]);

    printf("Please enter vector values: ");
    for (col = 0 ; col < 3 ; col++)
        scanf("%d", &vector[col]);

    for (row = 0 ; row < 3 ; row++) {
        result[row] = 0;
        for (col = 0 ; col < 3 ; col++)
            result[row] += matrix[row][col] * vector[col];
    }

    printf("The result is: \n");
    for (col = 0 ; col < 3 ; col++)
        printf("%d \n", result[col]);
}

```

2.

```
#include <stdio.h>
void main()
{
    char tic[10][10];
    int row, col, n;

    printf("Please enter a number: ");
    scanf("%d", &n);

    for (row = 0 ; row < 10 ; row++)
        for (col = 0 ; col < 10 ; col++)
            tic[row][col] = ' ';

    for (row = 1 ; row <= n ; row += 2)
        for (col = (n - row) / 2 ;
            col < row + (n - row) / 2 ; col++)
            tic[row / 2][col] = '*';

    for (row = 0 ; row < n / 2 + 1 ; row++) {
        for (col = 0 ; col < n ; col++)
            printf("%c", tic[row][col]);
        printf("\n");
    }
}
```

3.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int tic[3][3], row, col, winner = -1;

    printf("Please enter board position: \n");
    for (row = 0 ; row < 3 ; row++) {
        for (col = 0 ; col < 3 ; col++) {
            tic[row][col] = getch();
            printf("%c ", tic[row][col]);
        }
        printf("\n");
    }

    for (row = 0 ; row < 3 && winner == -1; row++)
        if (tic[row][0] == tic[row][1] &&
            tic[row][1] == tic[row][2])
            winner = tic[row][0];
}
```

```

else
    if (tic[0][row] == tic[1][row] &&
        tic[1][row] == tic[2][row])
        winner = tic[0][row];

if (winner == -1)
    if (tic[0][0] == tic[1][1] && tic[1][1] == tic[2][2] ||
        tic[0][2] == tic[1][1] && tic[1][1] == tic[2][0])
        winner = tic[1][1];

if (winner == -1)
    printf("No one wins! \n");
else
    printf("Player %c wins! \n", winner);
}

```

.14

אין לאתחל מערך במספר ערכים הגדול ממספר איבריו	.7 x	יש לתחום את הערכים המאתחלים בסוגריים מסולסלים	.2 x
	.8 ✓	יש להפריד את הערכים המאתחלים בפסיקים	.3 x
יש לתחום את הערכים המאתחלים בסוגריים מסולסלים	.10 x	יש לכתוב ערך מאתחל בין כל פסיק ופסיק (במילים אחרות, אין לדלג על ערכים מאתחלים)	.4 x
	.11 ✓		.5 ✓
	.12 ✓		.6 ✓

יש להפריד בין הסוגריים המסולסלים התוחמים את הערכים בכל שורה בפסיקים	.10 x	2, 4, 6, 8, 3, 6, 9, 12, 4, 8, 12, 16	.1 ✓
אין להוסיף סוגריים מסולסלים יתר על המידה (על צמד סוגריים מסולסלים אחד לתחום את כל איברי המערך, וניתן להוסיף עוד צמד לכל שורה)	.11 x	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	.2 ✓
על הסוגריים המסולסלים התוחמים את הערכים בכל שורה להימצא בין צמד נוסף של סוגריים מסולסלים	.12 x	0, 1, 1, 2, 4, 7, 13, 24, 44, 81	.3 ✓
על כל צמד של סוגריים מסולסלים נוספים (כלומר, פרט לאלו התוחמים את כל איברי המערך) לתחום שורת איברים בודדת במערך	.13 x	אין לאתחל מערך במספר ערכים הגדול ממספר איבריו	.4 x
0, 1, 0, 1, 1, 0, 1, 0, 0	.14 ✓	1.0, 2.0, 3.0, 0.0, 0.0	.5 ✓
0, 1, 1, 1, 1, 0, 0, 0, 0	.15 ✓	0, 0, 0, 0	.6 ✓
0, 1, 2, 6, 24, 120, 720, 0	.16 ✓	אין לאתחל מערך במספר ערכים הגדול ממספר איבריו (בשל אופרטור הפסיק, בשורה זו יש ניסיון להגדיר מערך חד-מימדי בן 4 איברים)	.7 x
אין להשמיט את גודל מימדו השני של המערך	.17 x	יש לתחום את הערכים המאתחלים בסוגריים מסולסלים	.8 x
אין להשמיט את גודל מימדו השני של המערך	.18 x	1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3	.9 ✓

.16

1. `int ri[3][4][4];`
2. `int ri[2][3][2];`
3. `int ri[2][3][5];`
4. `int ri[3][4][2];`
5. `int ri[3][3][3];`
6. `int ri[2][2][3];`

.17

(מיקומי האיברים מובאים מימין לשמאל)

- | | | |
|---|---|---|
| <ol style="list-style-type: none"> 1. <code>[2][2][2]</code>
<code>[2][0][3]</code>
<code>[1][0][1]</code>
<code>[0][0][0]</code>
<code>[1][1][0]</code> | <ol style="list-style-type: none"> 2. <code>[1][1][4]</code>
<code>[1][0][4]</code>
<code>[0][0][3]</code>
<code>[1][0][1]</code>
<code>[0][2][0]</code> | <ol style="list-style-type: none"> 3. <code>[2][2][1]</code>
<code>[2][0][1]</code>
<code>[0][0][0]</code>
<code>[0][1][0]</code>
<code>[1][2][0]</code> |
|---|---|---|

.18

1. `int ri[2][2][2] = { 2, 3, 5, 7, 11, 13, 17, 19 };`
2. `int ri[3][2][2] = { 0, 1, 2, 3, 6, 11, 20, 37 };`
3. `int ri[3][2][3] = { { 1, 2, 3 },`
 `{ 4, 5, 6 },`
 `{ 7, 8, 9 } };`
4. `int ri[2][2][3][2] = { 100, 200, 300, 200, 100 };`
5. `int ri[2][2][2][3] = { { {123}, {456} }, {789} };`

.19

.3 ✓	.1 x
.4 ✓	.2 ✓

1.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <stdio.h>
void PrintArray(int array[], int n)
{
    int i;

    for (i = 0 ; i < n ; i++)
        printf("%d ", array[i]);
}

```

2.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <stdio.h>
int InputArray(int array[], int n)
{
    int i, num, total = 0;

    for (i = 0 ; i < n ; i++) {
        scanf("%d", &num);
        array[i] = num;
        total += num;
    }

    return total;
}

```

3.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
void DoubleArray(int array1[], int array2[], int n)
{
    int i;

    for (i = 0 ; i < n ; i++)
        array2[i] = array1[i] * array1[i];
}

```

.3 ✓	.1 x
.4 x	.2 x

1.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <stdio.h>
void ReadValues(int array[][4], int n)
{
    int i, j;

    for (i = 0 ; i < n ; i++)
        for (j = 0 ; j < 4 ; j++)
            scanf("%d", &array[i][j]);
}

```

2.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
int FindBiggest(int array[][5], int n)
{
    int i, j, max;

    max = array[0][0];
    for (i = 0 ; i < n ; i++)
        for (j = 0 ; j < 5 ; j++)
            if (max < array[i][j])
                max = array[i][j];

    return max;
}

```

3.

```
/* note: this is not a complete program,  
 * because no main() function is defined.  
 */  
void SumArray(int source[][4], int target[], int n)  
{  
    int i, j, sum;  
  
    for (i = 0 ; i < 4 ; i++) {  
        sum = 0;  
        for (j = 0 ; j < n ; j++)  
            sum += source[j][i];  
        target[i] = sum;  
    }  
}
```

פרק 10

מחרוזות

שאלות

הקבוע המחרוזתי

1. הקף בעיגול את המשפטים הנכונים:

1. כל מלל נחשב לקבוע מחרוזתי.
2. הקבוע המחרוזתי "abc" זהה לתו 'abc'.
3. אל סוף מלל התחום ב-" תמיד מתווסף תו null ('\0').
4. ניתן להחליף כל מופע של התו '\0' בערך 0.
5. ניתן להחליף כל מופע של התו '\0' בתו '0'.
6. תו null מוצג על המסך כאשר נפלט הקבוע המחרוזתי.

אתחול מחרוזות

2. אחדות משורות ההגדרה שלפניך שגויות. סמן אותן והסבר מהן הטעויות:

1. `char hello[6] = { 'h', 'e', 'l', 'l', 'o', '\0' };`
2. `char aloha[10] = { "a", "l", "o", "h", "a", "\0" };`
3. `char howdy[10] = "howdy";`
4. `char hola[3] = "hola!";`
5. `char ahlan[6] = 'ahlan';`
6. `char kramer[] = "yo yo ma";`
7. `char meaning[];`
8. `char hi[] = { 'h', 'i', '\0' };`

3. כתוב את התוכניות הבאות:

1. הגדר מחרוזת ואתחל אותה בקבוע מחרוזתי כלשהו. התוכנית תפלוט את כל תווי המחרוזת שאינם אמות קריאה (אמות הקריאה הן האותיות 'a', 'e', 'i', 'o' ו-'u').

2. הגדר מחרוזת ואתחל אותה בקבוע מחרוזתי כלשהו. התוכנית תפלוט את תווי המחרוזת, כך שכל תו ייפלט פעם אחת יותר ממספר הפעמים שנפלט התו הקודם לו. התו הראשון ייפלט פעם אחת. למשל, המחרוזת "quack" תיפלט כ-"quaaaaccccckkkkk".

3. הגדר מחרוזת ואתחל אותה בקבוע מחרוזתי כלשהו. התוכנית תפלוט את תווי המחרוזת שהאינדקס שלהם הוא אחד מאיברי סדרת פיבונאצ'י.

4. כתוב את התוכניות הבאות:

1. הגדר מחרוזת ואתחל אותה בקבוע מחרוזתי כלשהו. התוכנית תקלוט תו מהמשתמש, ותפלוט את תווי המחרוזת, כך שבמקום התווים המופיעים באינדקסים הזוגיים ייפלט התו שהוון.

2. הגדר מחרוזת ואתחל אותה בקבוע מחרוזתי כלשהו. התוכנית תקלוט תו מהמשתמש ותמצא אם הוא מופיע במחרוזת, ואם כן – כמה פעמים.

3. הגדר מחרוזת ואתחל אותה בקבוע מחרוזתי כלשהו. התוכנית תקלוט שני מספרים טבעיים מהמשתמש (השני גדול מהראשון), המייצגים שני אינדקסים של תווים במחרוזת, ותהפוך את סדר התווים המופיעים בין שני האינדקסים. תווים אלו ייפלטו, בסדר החדש בו הם מופיעים במחרוזת.

המחרוזות והמשתמש

פלט מחרוזות / קליטת מחרוזות / הפונקציה gets

5. נתונים שלושה קטעי תוכניות:

```
1. char str[20];  
   printf("Please enter a string: ");  
   scanf("%s", str);
```

```
2. char str[20];  
   printf("Please enter a string: ");  
   scanf("%s", &str);
```

```
3. char str[20];  
   printf("Please enter a string: ");  
   gets(str);
```

משתמש התוכנית הריץ את שלושת קטעי התוכנית, ובכל פעם שהתבקש הזין את הקלט הבא:

Original String

מה תכיל המחרוזת str בסוף כל אחד מקטעי התוכניות הללו?

6. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מחרוזות בת 15 תווים או פחות, ופולטת את תויה, תוך הפיכת האותיות הגדולות לקטנות והשמטת התווים שאינם אותיות. למשל, עבור המחרוזת "VB!Sck5" ייפלט המלל הבא:

vbsck

2. תוכנית הקולטת מהמשתמש מחרוזות בת 30 תווים או פחות, תו למציאה ותו להחלפה. התוכנית תחליף כל הופעה של תו המציאה במחרוזת בתו ההחלפה, ולבסוף, תפלוט את המחרוזת. למשל, עבור המחרוזת "eating and feasting", תו המציאה 'a' ותו ההחלפה 'e', ייפלט המלל הבא:

eeting end feasting

3. תוכנית הקולטת מהמשתמש מחרוזות בת 20 תווים או פחות, ופולטת את מספר התווים שהוזנו (לא כולל תו ה-\0').

7. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מחרוזת בת 10 תווים או פחות, ופולטת את תווייה באלכסון. למשל, עבור המחרוזת "zealot" ייפלט המלל הבא:

```
z
 e
  a
   l
    o
     t
```

2. תוכנית הקולטת מהמשתמש מחרוזת בת 10 תווים או פחות, ופולטת אותה בצורת משולש ישר-זווית. למשל, עבור המחרוזת "iguana" ייפלט המלל הבא:

```
i
ig
igu
igua
iguan
iguana
```

3. תוכנית הקולטת מהמשתמש מחרוזת בת מספר אי-זוגי של תווים הקטן מ-20, ופולטת אותה בצורת משולש שווה-שוקיים. למשל, עבור המחרוזת "reptile" ייפלט המלל הבא:

```
reptile
 eptil
  pti
   t
```

8. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות, ופולטת את מספר המילים שבה (מילה מופרדת מרעותה על ידי תו רווח אחד או יותר).

2. תוכנית הקולטת מהמשתמש מחרוזת בת 25 תווים או פחות, ופולטת את תווייה, תוך השמטת התווים הזחים המופיעים שלוש פעמים או יותר בסמוך זה לזה. למשל, עבור המחרוזת "abbbccddeeeebcccd" ייפלט המלל הבא:

```
accddb
```

3. תוכנית הקולטת מהמשתמש מחרוזת המייצגת מספר טבעי, ומציבה מספר זה במשתנה מסוג int. לבסוף, ייפלט ערך משתנה זה.

9. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מחרוזת בת 30 תווים או פחות, המכילה, בין היתר, סוגריים מסולסלים פותחים וסוגרים. התוכנית תפלוט הודעה מתאימה אם מספר הסוגריים המסולסלים תואם את חוקי שפת C, הקובעים כי מספר הסוגריים המסולסלים הפותחים והסוגרים חייב להיות שווה.
2. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות המכילה הערה בשפת C, ופולטת את התווים המופיעים בתוכה. הנח כי במחרוזת מופיעה הערה אחת בלבד, הנפתחת בתווים /* ומסתיימת בתווים */.
3. תוכנית הקולטת מהמשתמש מספר טבעי מהמשתמש ומציבה אותו במחרוזת. לבסוף, תיפלט המחרוזת מספר פעמים השווה למספר הספרות שבמספר שהוזן.

10. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות, ופולטת את תויה, כך שלא יופיעו שנים או יותר תווים זהים בסמוך זה לזה. למשל, עבור המחרוזת "teentttaclllee" ייפלט המלל הבא :
tentacle
2. תוכנית הקולטת מהמשתמש מחרוזת בת 10 תווים ותו. התוכנית תפלוט את תווי המחרוזת, כך שהתו יופיע בין כל שני תווים במחרוזת. בכל פעם, יופיע תו זה פעמיים יותר מאשר בפעם הקודמת שהופיע. למשל, עבור המחרוזת "cipher" והתו '-' ייפלט המלל הבא :
c-i---p-----h-----e-----r
3. תוכנית הקולטת מהמשתמש שתי מחרוזות בנות 20 תווים או פחות ופולטת את התווים המופיעים בשתייהן. למשל, עבור המחרוזות "mush hashor" ו-"suspenders" ייפלט המלל הבא :
u s r

11. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות ופולטת את תויה באופן הבא : אם מופיע במחרוזת התו '-' תתבצע ירידה לשורה חדשה. אם מופיע במחרוזת התו '.' ייפלט התו '.'. אם מופיע במחרוזת התו ' ' ייפלט התו ' ' ,'. אם מופיע במחרוזת התו ',' ייפלט התו '-'. שאר התווים ייפלטו כרגיל.
2. תוכנית הקולטת מהמשתמש שתי מחרוזות בנות 15 תווים או פחות, ומחליפה בין תווי המחרוזות שהאינדקס שלהם זוגי. לבסוף, ייפלטו שתי המחרוזות. למשל, עבור המחרוזות "duck" ו-"shiver" ייפלט המלל הבא :

suik
dhcver

3. תוכנית הקולטת מהמשתמש שתי מחרוזות בנות 20 תווים או פחות, כאשר השנייה קצרה מן הראשונה. התוכנית תפלוט הודעה מתאימה אם המחרוזות השנייה מופיעה בראשונה. למשל, עבור המחרוזות "alligator" ו-"gat" ייפלט המלל הבא:

"gat" was found in "alligator"!

12. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש שתי מחרוזות בנות 20 תווים או פחות, ופולטת את תויהן בשתי-וערב; כלומר, את התו הראשון מהמחרוזת הראשונה, אחריו את התו הראשון מהמחרוזת השנייה, אחריו את התו השני מהמחרוזת הראשונה, אחריו את התו השני מהמחרוזת השנייה, וכדומה. אם אחת מהמחרוזות ארוכה מהשנייה, ייפלטו תוויה שלא נפלטו בשתי-וערב לאחר התווים שנפלטו. למשל, עבור המחרוזות "May" ו-"January" ייפלט המלל הבא:

MJaaynuary

2. תוכנית הקולטת מהמשתמש מחרוזת בת 30 תווים או פחות, המורכבת מאותיות לעזיות ומרווחים, ומייצגת שם מלא של אדם. התוכנית תפלוט את ראשי התיבות שלו. למשל, עבור המחרוזת "Shvirgen Van Der Hogen" (ייתכן שיותר מתו רווח אחד יפריד בין שתי מילים) ייפלט המלל הבא:

SVDH

3. תוכנית הקולטת מהמשתמש מחרוזת בת 10 תווים או פחות, ופולטת כמה פעמים בה הופיע כל תו. למשל, עבור המחרוזת "grammar" ייפלט המלל הבא:

g - 1
r - 2
a - 2
m - 2

13. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות ומספר המציין אינדקס במחרוזת. התוכנית תוציא את התו שבאינדקס זה מהמחרוזת (על ידי העברת כל התווים הבאים אחריו אחורנית), ולבסוף, תפלוט אותה.

2. תוכנית הקולטת מהמשתמש מחרוזת בת 15 תווים או פחות ומוציאה ממנה את כל הספרות. לבסוף, תפלוט התוכנית את המחרוזת.

3. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות ומכפילה את כל התווים שבה. לבסוף, תפלוט התוכנית את המחרוזת. למשל, עבור המחרוזת "aquarium" ייפלט המלל הבא:

aaqquaarriuumm

פונקציות ספריה לטיפול במחרוזות

מציאת אורך מחרוזת

14. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות ותו למציאה. התוכנית תפלוט את אינדקס האיבר בו מופיע תו זה, הקרוב ביותר לסוף המחרוזת. אם התו לא מופיע כלל במחרוזת, תיפלט הודעה מתאימה. למשל, עבור המחרוזת "jogging" והתו 'g' ייפלט האינדקס 6.
2. תוכנית הקולטת מהמשתמש מחרוזת בת 15 תווים או פחות ומציגה אותה כמחרוזת פלינדרומית, על ידי הדפסת כל תוויה מהסוף להתחלה ומהתחלה לסוף. למשל, עבור המחרוזת "elephant" ייפלט המלל הבא:
tnahpeleelephant
3. תוכנית הקולטת מהמשתמש מחרוזת בת 40 תווים או פחות ומוצאת האם היא מחרוזת פלינדרומית, תוך התעלמות מהתווים שאינם אותיות לועזיות. למשל, המחרוזות "a man, a plan, a canal - panama!" , "a madam, I'm adam" ו-"ab.b - a" ייחשבו למחרוזות פלינדרומיות.

העתקת מחרוזת

15. לפניך תוכנית האמורה לפלוט מחרוזת כלשהי על המסך.

- ❖ התוכנית שגויה בשל היעדרה של שורה אחת. הוסף שורה זו לתוכנית.
- ❖ מצא את תוכן המחרוזת לאחר הרצת התוכנית.

```
#include <stdio.h>
void main()
{
    char string[20] = "????";

    strcpy(string, "one line is missing");
    strcpy(string, "find it");
    strcpy(string, "!");
}
```

16. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש עשר מחרוזות בנות 20 תווים או פחות ופולטת את המחרוזת הקצרה ביותר שהוזנה.
2. תוכנית הקולטת מהמשתמש שלוש מחרוזות בנות 15 תווים או פחות ומספר טבעי n . התוכנית תחליף בין כל המחרוזות n פעמים, ולאחר כל החלפה, תפלוט אותן. בכל החלפה תועבר המחרוזת הראשונה לשנייה, השנייה לשלישית והשלישית לראשונה.
3. תוכנית הקולטת מהמשתמש מחרוזת בת 20 תווים או פחות, ומעתיקה אותה למחרוזת נוספת, מבלי להשתמש בפונקציה `strcpy`.

הוספת מחרוזות

17. כל אחד מקטעי התוכניות הבאים שגוי או מסוכן. הסבר מדוע.

1.

```
char first[] = "november", second[10] = " rain";  
strcat(first, second);
```
2.

```
char target[10] = "dan";  
strcat("gerous", target);
```
3.

```
char str1[5], str2[] = "ab";  
strcat(str1, str2);
```
4.

```
char name[3] = "u";  
strcat(name, '2');
```

18. כתוב את התוכניות הבאות :

1. תוכנית הקולטת מהמשתמש שתי מחרוזות בנות 10 תווים או פחות, ומוסיפה כל אחת אל השנייה. לבסוף, ייפלטו שתי המחרוזות. למשל, עבור המחרוזות "Library" ו-"Function" ייפלט המלל הבא :

```
LibraryFunction  
FunctionLibrary
```

2. תוכנית הקולטת מהמשתמש שלוש מחרוזות בנות 15 תווים או פחות, המכילות את שמו הפרטי, את שם משפחתו ואת כתובתו. התוכנית תעתיק נתונים אלו, יחד עם קבועים מחרוזתיים המסבירים אותם, אל מחרוזת גדולה יותר, ותפלוט אותה על המסך. למשל, עבור המחרוזות "Homer", "Simpson" ו-"Nuclear Plant" ייפלט המלל הבא :

```
First name: Homer  
Last name: Simpson  
Address: Nuclear Plant
```

3. תוכנית הקולטת מהמשתמש מחרוזת בת 10 תווים או פחות ובאמצעות strcpy מעתיקה אותה n פעמים אל מחרוזת שנייה, בת 50 תווים, כאשר n הוא מספר שמזין המשתמש. לבסוף, התוכנית תפלוט את המחרוזת. לפני ההעתקות, ודא כי המחרוזת הארוכה גדולה דיה כדי להכיל את איברי המחרוזת הקצרה n פעמים.

השוואת מחרוזות

19. מצא איזה ערך (חיובי, שלילי או אפס) יוצב במשתנה i לאחר ביצוע כל אחד מהמשפטים הבאים:

1. `i = strcmp("mars", "saturn");`
2. `i = strcmp("brick", "break");`
3. `i = strcmp("strcat(a, b)", "strcmp(b, a)");`
4. `i = strcmp("rush", "rush");`
5. `i = strcmp("drop", "DROP");`
6. `i = strcmp("Stark", "naked");`
7. `i = strcmp("language", "Prolog");`
8. `i = strcmp("295", "295.3");`

20. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש שתי מחרוזות בנות 20 תווים או פחות שוב ושוב, עד ששתיהן זהות זו לזו.
2. תוכנית הקולטת מהמשתמש עשר מחרוזות בנות 15 תווים או פחות, המורכבות מאותיות קטנות בלבד, ופולטת מי מהן מופיעה ראשונה במילון ומי אחרונה.
3. כתוב את תוכנית 1 מבלי להשתמש בפונקציה strcmp.

מערכי מחרוזות

21. כתוב את התוכניות הבאות:

1. תוכנית הקולטת מהמשתמש חמש מחרוזות בנות 10 תווים או פחות, ומאחסנת אותן במערך מחרוזות מתאים. התוכנית תפלוט כל מחרוזת מספר פעמים השווה למספר התווים שבה.
2. תוכנית הקולטת מהמשתמש חמש מחרוזות בנות 15 תווים או פחות, ומאחסנת אותן במערך מחרוזות מתאים. התוכנית תהפוך את כל האותיות הלועזיות הגדולות במחרוזות לאותיות לועזיות קטנות, ותפלוט את המחרוזות בסדר הפוך לסדר בו הוזנו.
3. תוכנית הקולטת מהמשתמש עשר מחרוזות בנות 15 תווים או פחות, ומאחסנת אותן במערך מחרוזות מתאים. התוכנית תמייך את המחרוזות בסדר עולה (על ידי מיון בועות והפונקציה `strcmp`) ותפלוט אותן.

שליחת מחרוזות לפונקציות

שליחת מחרוזת

22. כתוב את הפונקציות הבאות:

1. פונקציה המקבלת כפרמטר מחרוזת ומחזירה את מספר האותיות הלועזיות שבה.
2. פונקציה המקבלת כפרמטר מחרוזת ותו, ומחזירה את מספר הפעמים שהתו מופיע במחרוזת.
3. פונקציה המקבלת כפרמטר ארבע מחרוזות, ומחזירה את אורך המחרוזת הקצרה מביניהן.

23. כתוב את הפונקציות הבאות:

1. פונקציה המקבלת כפרמטר מחרוזת ומחזירה את מספר התווים שבה עד לתו `'\0'`, ולא כולל אותו (פונקציה זו פועלת, למעשה, כמו פונקציית הספרייה `strlen`).
2. פונקציה המקבלת כפרמטר שתי מחרוזות ומעתיקה את המחרוזת השנייה לתחילת המחרוזת הראשונה (פונקציה זו פועלת, למעשה, כמו פונקציית הספרייה `strcpy`).

3. פונקציה המקבלת כפרמטר שתי מחרוזות ומעתיקה את המחרוזת השנייה לסוף המחרוזת הראשונה (פונקציה זו פועלת, למעשה, כמו פונקציית הספריה strcat).

4. פונקציה המקבלת כפרמטר שתי מחרוזות ומוצאת, בדומה לפונקציה strcmp, מי מהן קודמת לשנייה בסדר לקסיקוגרפי.

שליחת מערך מחרוזות

24. כתוב את הפונקציות הבאות:

1. פונקציה המקבלת כפרמטרים מערך מחרוזות בנות 20 תווים או פחות ומספר n הקובע את מספר המחרוזות בו. התוכנית תקלוט מחרוזות מהמשתמש ותאחסן אותן במערך, ולאחר-מכן תחזיר את סכום אורכי המחרוזות שהוזנו.

2. פונקציה המקבלת כפרמטרים מערך מחרוזות בנות 10 תווים או פחות ומספר n הקובע את מספר המחרוזות בו, ומחזירה את אינדקס המחרוזת הקודמת לכל יתר המחרוזות בסדר לקסיקוגרפי.

3. פונקציה המקבלת כפרמטרים מערך מחרוזות בנות 20 תווים או פחות ומספר n הקובע את מספר המחרוזות בו, וממיינת את המחרוזות על-פי אורכן (כך שהמחרוזת הקצרה ביותר תופיע בתחילת המערך והמחרוזת הארוכה ביותר תופיע בסופו).

תשובות

.1

.4 ✓	.1 x
.5 x	.2 x
.6 x	.3 ✓

.2

.5 x	.1 ✓
את הקבוע המחרוזתי יש לתחום בגרשיים כפולים (") ולא בגרשיים בודדים (!)	.2 x
.6 ✓	את התווים המשמשים כערכים מאתחלים יש לתחום בגרשיים בודדים (!)
.7 x	.3 ✓
כאשר המחרוזת אינה מאותחלת חובה לציין את גודלה	.4 x
.8 ✓	המחרוזת hola אינה די גדולה כדי להכיל את כל תווי הקבוע המחרוזתי

.3

1.

```
#include <stdio.h>
void main()
{
    char str[] = "Breaking all the rules";
    int i;

    for (i = 0 ; str[i] != '\0' ; i++)
        if (str[i] != 'a' && str[i] != 'e' && str[i] != 'i' &&
            str[i] != 'o' && str[i] != 'u')
            printf("%c", str[i]);
}
```

2.

```
#include <stdio.h>
void main()
{
```

```

char string[] = "freaking COOL";
int i, j;

for (i = 0 ; string[i] != '\0' ; i++)
    for (j = 0 ; j < i + 1 ; j++)
        printf("%c", string[i]);
}

```

3.

```

#include <stdio.h>
void main()
{
    char str[] = "Hovering over mars";
    int a1 = 1, a2 = 1, temp, i;

    if (str[0] != '\0')
        printf("%c", str[0]);
    for (i = 0 ; str[i] != '\0' ; i++)
        if (i == a2) {
            printf("%c", str[i]);
            temp = a2;
            a2 += a1;
            a1 = temp;
        }
}

```

.4

1.

```

#include <stdio.h>
void main()
{
    char string[] = "AlgorithmshtiroglA", ch;
    int i;

    printf("Please enter a char: ");
    scanf("%c", &ch);

    for (i = 0 ; string[i] != '\0' ; i++)
        if (i % 2 == 0)
            printf("%c", ch);
        else
            printf("%c", string[i]);
}

```

2.

```
#include <stdio.h>
void main()
{
    char string[] = "Octopuses vs Dogs", ch;
    int i, n = 0;

    printf("Please enter a char: ");
    scanf("%c", &ch);

    for (i = 0 ; string[i] != '\0' ; i++)
        if (string[i] == ch)
            n++;

    if (n == 0)
        printf("'%' wasn't found in the string! \n", ch);
    else
        printf("'%' was found in the string %d time(s). \n", n);
}
```

3.

```
#include <stdio.h>
void main()
{
    char str[] = "string manipulation";
    int i, from, to, temp;

    printf("Please enter two indexes: ");
    scanf("%d %d", &from, &to);

    for (i = 0 ; i < (to - from) / 2 ; i++) {
        temp = str[from + i];
        str[from + i] = str[to - i];
        str[to - i] = temp;
    }

    for (i = from ; i <= to ; i++)
        printf("%c", str[i]);
    printf("\n");
}
```

-
1. Original
 2. Original
 3. Original String
-

1.

```
#include <stdio.h>
void main()
{
    char str[16];
    int i;

    printf("Please enter a string: ");
    gets(str);

    for (i = 0 ; str[i] != '\0' ; i++)
        if (str[i] >= 'a' && str[i] <= 'z')
            printf("%c", str[i]);
        else if (str[i] >= 'A' && str[i] <= 'Z')
            printf("%c", str[i] + 'a' - 'A');
}
```

2.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char str[31], find, replace;
    int i;

    printf("Please enter a string: ");
    gets(str);
    printf("\nPlease enter a char to find: ");
    find = getch();
    putchar(find);
    printf("\nPlease enter a char to replace: ");
    replace = getch();
    putchar(replace);
}
```

```

for (i = 0 ; str[i] != '\0' ; i++)
    if (str[i] == find)
        str[i] = replace;

printf("\n%s \n", str);
}

```

3.

```

#include <stdio.h>
void main()
{
    char str[21];
    int length = 0;

    printf("Please enter a string: ");
    gets(str);

    while (str[length] != '\0')
        length++;

    printf("Length of \"%s\": %d \n", str, length);
}

```

.7

1.

```

#include <stdio.h>
void main()
{
    char str[11];
    int i, space;

    printf("Please enter a string: \n");
    gets(str);

    for (i = 0 ; str[i] != '\0' ; i++) {
        for (space = 0 ; space < i ; space++)
            printf(" ");
        printf("%c \n", str[i]);
    }
}

```

2.

```
#include <stdio.h>
void main()
{
    char str[11];
    int i, j;

    printf("Please enter a string: \n");
    gets(str);

    for (i = 0 ; str[i] != '\0' ; i++) {
        for (j = 0 ; j <= i ; j++)
            printf("%c", str[j]);
        printf("\n");
    }
}
```

3.

```
#include <stdio.h>
void main()
{
    char str[20];
    int i, start, end = 0;

    printf("Please enter a string: ");
    gets(str);

    while (str[end] != '\0')
        end++;
    end--;

    for (start = 0 ; start <= end ; start++) {
        for (i = 0 ; i < start ; i++)
            printf(" ");
        for (i = start ; i <= end ; i++)
            printf("%c", str[i]);
        end--;
        printf("\n");
    }
}
```

1.

```

#include <stdio.h>
void main()
{
    char str[21];
    int i = 0, count = 0;

    printf("Please enter a string: ");
    gets(str);

    while (str[i] != '\0') {
        while (str[i] == ' ')
            i++;
        if (str[i] != '\0')
            count++;
        while (str[i] != ' ' && str[i] != '\0')
            i++;
    }

    printf("Words count: %d \n", count);
}

```

2.

```

#include <stdio.h>
void main()
{
    char str[26];
    int i = 0, j, count;

    printf("Please enter a string: ");
    gets(str);

    while (str[i] != '\0') {
        count = 1;
        while (str[i + count] == str[i])
            count++;
        if (count <= 2)
            for (j = 0 ; j < count ; j++)
                printf("%c", str[i]);
        i += count;
    }
}

```

3.

```
#include <stdio.h>
void main()
{
    char str[6];
    int i, num = 0;

    printf("Please enter a number: ");
    gets(str);

    for (i = 0 ; str[i] != '\0' ; i++)
        num = num * 10 + str[i] - '0';
    printf("Number is: %d \n", num);
}
```

.9

1.

```
#include <stdio.h>
void main()
{
    char str[31];
    int i, count = 0;

    printf("Please enter a string: ");
    gets(str);

    for (i = 0 ; str[i] != '\0' ; i++)
        if (str[i] == '{')
            count++;
        else if (str[i] == '}')
            count--;

    if (count == 0)
        printf("The number of { and } matches C grammar! \n");
    else
        printf("The number of { and } doesn't match C
        grammar! \n");
}
```

2.

```
#include <stdio.h>
void main()
{
    char str[21], comment[17];
    int i, j, found = 0;
```

```

printf("Please enter a string: ");
gets(str);

for (i = 0 ; str[i] != '\0' && found == 0 ; i++)
    if (str[i] == '/' && str[i + 1] == '*') {
        for (j = i + 2 ;
            str[j] != '*' || str[j + 1] != '/' ; j++)
            comment[j - i - 2] = str[j];
        comment[j - i - 2] = '\0';
        found = 1;
    }

printf("Comment is: \"%s\"", comment);
}

```

3.

```

#include <stdio.h>
void main()
{
    char str[11];
    int num, temp, count = 0;

    printf("Please enter a number: ");
    scanf("%d", &num);

    for (temp = num ; temp > 0 ; temp /= 10)
        count++;

    str[count] = '\0';
    while (num > 0) {
        count--;
        str[count] = num % 10 + '0';
        num /= 10;
    }

    for (temp = 0 ; str[temp] != '\0' ; temp++)
        printf("%s \n", str);
}

```

1.

```

#include <stdio.h>
void main()
{
    char str[21];
    int i = 0, count;

    printf("Please enter a string: ");
    gets(str);

    while (str[i] != '\0') {
        printf("%c", str[i]);
        count = 1;
        while (str[i + count] == str[i])
            count++;
        i += count;
    }
}

```

2.

```

#include <stdio.h>
void main()
{
    char str[11], ch;
    int i, j, count = 1;

    printf("Please enter a string: ");
    gets(str);
    printf("Please enter a char: ");
    scanf("%c", &ch);

    for (i = 0 ; str[i] != '\0' ; i++) {
        printf("%c", str[i]);
        if (str[i + 1] != '\0') {
            for (j = 0 ; j < count ; j++)
                printf("%c", ch);
            count += 2;
        }
    }
}

```

3.

```
#include <stdio.h>
void main()
{
    char str1[21], str2[21], chars_found[20] = { 0 };
    int i, j, count = 0, found;

    printf("Please enter two strings: \n");
    gets(str1);
    gets(str2);

    for (i = 0 ; str1[i] != '\0' ; i++) {
        j = 0;

        /* check if str1[i] appears somewhere in chars_found */
        while (j < count && str1[i] != chars_found[j])
            j++;

        if (j == count) {
            found = 0;
            for (j = 0 ; str2[j] != '\0' && found == 0 ; j++)
                if (str1[i] == str2[j]) {
                    chars_found[count] = str1[i];
                    count++;
                    found = 1;
                }
        }
    }

    for (i = 0 ; chars_found[i] != 0 ; i++)
        printf("%c ", chars_found[i]);
}
```

.11

1.

```
#include <stdio.h>
void main()
{
    char str[21];
    int i;

    printf("Please enter a string: ");
    gets(str);

    printf("Special output of the string: \n");
}
```

```

for (i = 0 ; str[i] != '\0' ; i++)
    if (str[i] == '-')
        printf("\n");
    else if (str[i] == '.')
        printf(" ");
    else if (str[i] == ' ')
        printf(",");
    else if (str[i] == ',')
        printf("-");
    else printf("%c", str[i]);
}

```

2.

```

#include <stdio.h>
void main()
{
    char str1[16], str2[16], temp;
    int i;

    printf("Please enter two strings: \n");
    gets(str1);
    gets(str2);

    for (i = 0 ; str1[i] != '\0' && str2[i] != '\0' ; i++)
        if (i % 2 == 0) {
            temp = str1[i];
            str1[i] = str2[i];
            str2[i] = temp;
        }

    printf("Strings after the replace: \n%s\n%s\n", str1, str2);
}

```

3.

```

#include <stdio.h>
void main()
{
    char str1[21], str2[21];
    int i, j = 0;

    printf("Please enter two strings: \n");
    gets(str1);
    gets(str2);
}

```

```

for (i = 0 ; str1[i] != '\0' && str2[j] != '\0' ; i++) {
    if (str1[i] == str2[j])
        j++;

    else {
        if (str1[i] == str2[0])
            i--;
        j = 0;
    }
}

if (str2[j] == '\0')
    printf("\"%s\" was found in \"%s\"! \n", str2, str1);
else
    printf("\"%s\" wasn't found in \"%s\"! \n", str2, str1);
}

```

.12

1.

```

#include <stdio.h>
void main()
{
    char str1[21], str2[21];
    int i, j;

    printf("Please enter two strings: \n");
    gets(str1);
    gets(str2);

    for (i = 0 ; str1[i] != '\0' && str2[i] != '\0' ; i++)
        printf("%c%c", str1[i], str2[i]);

    j = i;
    for ( ; str1[i] != '\0' ; i++)
        printf("%c", str1[i]);

    for ( ; str2[j] != '\0' ; j++)
        printf("%c", str2[j]);
}

```

2.

```
#include <stdio.h>
void main()
{
    char str[31];
    int i = 0;

    printf("Please enter a string: \n");
    gets(str);

    while (str[i] != '\0') {
        while (str[i] == ' ')
            i++;
        printf("%c", str[i]);

        while (str[i] != ' ' && str[i] != '\0')
            i++;
    }
}
```

3.

```
#include <stdio.h>
void main()
{
    char str[11];
    int counts[10][2] = { 0 }, i, j, unique = 0, found;

    printf("Please enter a string: ");
    gets(str);

    for (i = 0 ; str[i] != '\0' ; i++) {
        found = 0;
        for (j = 0 ; j < unique && found == 0 ; j++)
            if (str[i] == counts[j][0]) {
                counts[j][1]++;
                found = 1;
            }
        if (found == 0) {
            counts[unique][0] = str[i];
            counts[unique][1]++;
            unique++;
        }
    }
    for (i = 0 ; i < unique ; i++)
        printf("%c - %d \n", counts[i][0], counts[i][1]);
}
```

1.

```
#include <stdio.h>
void main()
{
    char str[21];
    int index;

    printf("Please enter a string: \n");
    gets(str);
    printf("Please enter an index: ");
    scanf("%d", &index);

    for ( ; str[index] != '\0' ; index++)
        str[index] = str[index + 1];
    printf("%s", str);
}
```

2.

```
#include <stdio.h>
void main()
{
    char str[16];
    int i = 0, j;

    printf("Please enter a string: ");
    gets(str);

    while (str[i] != '\0')
        if (str[i] >= '0' && str[i] <= '9')
            for (j = i ; str[j] != '\0' ; j++)
                str[j] = str[j + 1];
        else
            i++;

    printf("%s \n", str);
}
```

3.

```
#include <stdio.h>
void main()
{
    char str[41];
    int i, len = 0;
```

```

printf("Please enter a string: ");
gets(str);

while (str[len] != '\0')
    len++;

str[len * 2] = '\0';
for (i = len - 1 ; i >= 0 ; i--) {
    str[i + len] = str[i];
    str[i + len - 1] = str[i];
    len--;
}

printf("%s \n", str);
}

```

.14

1.

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
void main()
{
    char str[21], ch;
    int i, found = 0;

    printf("Please enter a string: ");
    gets(str);
    printf("Please enter a char to find: ");
    ch = getch();
    putchar(ch);

    for (i = strlen(str) - 1 ; i >= 0 && found == 0; i--)
        if (str[i] == ch) {
            printf("\nThe char was found in index %d \n", i);
            found = 1;
        }

    if (found == 0)
        printf("\nThe char wasn't found! \n");
}

```

2.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str[16];
    int i;

    printf("Please enter a string: ");
    gets(str);

    for (i = strlen(str) - 1 ; i >= 0 ; i--)
        printf("%c", str[i]);
    for (i = 0 ; str[i] != '\0' ; i++)
        printf("%c", str[i]);
}
```

3.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str[41];
    int from, to, is_palin = 1;

    printf("Please enter a string: ");
    gets(str);
    to = strlen(str) - 1;

    for (from = 0 ; from < to && is_palin == 1 ; from++) {
        while ((str[from] < 'a' || str[from] > 'z')
            && (str[from] < 'A' || str[from] > 'Z')
            && str[from] != '\0' && from < to)
            from++;
        while ((str[to] < 'a' || str[to] > 'z')
            && (str[to] < 'A' || str[to] > 'Z')
            && to > from)
            to--;

        if (str[from] != str[to])
            is_palin = 0;
        to--;
    }

    if (is_palin == 1)
        printf("\'%s\' is a palindromic string! \n", str);
}
```

```

else
    printf("\'%s\' is not a palindromic string! \n", str);
}

```

.15

– השורה החסרה

```
#include <string.h>
```

תוכן המחרוזת –

!	\0	n	d		i	t	\0		i	s		m	i	s	s	i	n	g	\0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

.16

1.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char str[21], shortest[21];
    int i;

    printf("Please enter ten strings: \n");

    for (i = 0 ; i < 10 ; i++) {
        gets(str);
        if (i == 0 || strlen(str) < strlen(shortest))
            strcpy(shortest, str);
    }

    printf("The shortest string is: %s \n", shortest);
}

```

2.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char str1[16], str2[16], str3[16], temp[16];
    int n;

    printf("Please enter three strings: \n");
    gets(str1);

```

```

gets(str2);
gets(str3);
printf("Please enter a number: ");
scanf("%d", &n);

for ( ; n > 0 ; n--) {
    strcpy(temp, str3);
    strcpy(str3, str2);
    strcpy(str2, str1);
    strcpy(str1, temp);

    printf("\n%s \n%s \n%s \n", str1, str2, str3);
}
}

```

3.

```

#include <stdio.h>
void main()
{
    char source[21], target[21];
    int i;

    printf("Please enter a string: ");
    gets(source);

    for (i = 0 ; source[i] != '\0' ; i++)
        target[i] = source[i];
    target[i] = '\0';

    printf("The string you've entered: %s \n", target);
}

```

.17

<p>3. המחרוזת str1 לא אותחלה, ומכיון שאיבריה מכילים ערכים בלתי-ידועים, הפונקציה strcat לא תמצא כהלכה את סופה</p>	<p>1. מחרוזת היעד (first) אינה די גדולה מספיק כדי להכיל גם את איברי מחרוזת המקור (second)</p>
<p>4. אל מחרוזת היעד יש להוסיף מחרוזת, ולא תו</p>	<p>2. הקבוע המחרוזתי ("gerous") צריך להישלח כפרמטר שני אל strcat, ואילו מחרוזת היעד (target) צריכה להישלח כפרמטר ראשון</p>

1.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char str1[21], str2[21], temp[11];

    printf("Please enter two strings: \n");
    gets(str1);
    gets(str2);

    strcpy(temp, str1);
    strcat(str1, str2);
    strcat(str2, temp);

    printf("After concatenating: \n%s \n%s \n", str1, str2);
}

```

2.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char first_name[16], last_name[16], address[16],
        str[80];

    printf("Please enter your first name: ");
    gets(first_name);
    printf("Please enter your last name: ");
    gets(last_name);
    printf("Please enter your address: ");
    gets(address);

    strcpy(str, "First name: ");
    strcat(str, first_name);
    strcat(str, "\nLast name: ");
    strcat(str, last_name);
    strcat(str, "\nAddress: ");
    strcat(str, address);

    printf("%s \n", str);
}

```

3.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str[51] = "", short_str[11];
    int n;

    printf("Please enter a string: ");
    gets(short_str);

    do {
        printf("Please enter a number: ");
        scanf("%d", &n);
    } while (strlen(short_str) * n > 50);

    while (n > 0) {
        strcat(str, short_str);
        n--;
    }

    printf("String after concatenating: %s \n", str);
}
```

.19

(>) חיובי .5	(<) שלילי .1
(<) שלילי .6	(>) חיובי .2
(>) חיובי .7	(<) שלילי .3
(<) שלילי .8	(=) אפס .4

.20

1.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str1[21], str2[21];

    do {
        printf("Please enter two strings: \n");
        gets(str1);
        gets(str2);
    }
```

```
    } while (strcmp(str1, str2) != 0);  
}
```

2.

```
#include <stdio.h>  
#include <string.h>  
void main()  
{  
    char str[16], first_str[16], last_str[16];  
    int i;  
  
    for (i = 0 ; i < 10 ; i++) {  
        printf("Please enter a string: ");  
        gets(str);  
        if (i == 0 || strcmp(str, first_str) < 0)  
            strcpy(first_str, str);  
        if (i == 0 || strcmp(str, last_str) > 0)  
            strcpy(last_str, str);  
    }  
  
    printf("First string: %s \n", first_str);  
    printf("Last string: %s \n", last_str);  
}
```

3.

```
#include <stdio.h>  
#include <string.h>  
void main()  
{  
    char str1[21], str2[21];  
    int i;  
  
    do {  
        printf("Please enter two strings: \n");  
        gets(str1);  
        gets(str2);  
  
        i = 0;  
        while (str1[i] != '\0' && str2[i] != '\0' &&  
            str1[i] == str2[i])  
            i++;  
    } while (str1[i] - str2[i] != 0);  
}
```

1.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char strings[5][11];
    int i, j, len;

    printf("Please enter five strings: \n");
    for (i = 0 ; i < 5 ; i++)
        gets(strings[i]);

    for (i = 0 ; i < 5 ; i++) {
        len = strlen(strings[i]);
        for (j = 0 ; j < len ; j++)
            printf("%s \n", strings[i]);
    }
}

```

2.

```

#include <stdio.h>
#include <string.h>
void main()
{
    char strings[5][16];
    int i, j, len;

    printf("Please enter five strings: \n");
    for (i = 0 ; i < 5 ; i++)
        gets(strings[i]);

    for (i = 4 ; i >= 0 ; i--) {
        len = strlen(strings[i]);
        for (j = 0 ; j < len ; j++)
            if (strings[i][j] >= 'A' && strings[i][j] <= 'Z')
                strings[i][j] += 'a' - 'A';
        printf("%s \n", strings[i]);
    }
}

```

3.

```
#include <stdio.h>
#include <string.h>
void main()
{
    char strings[10][16], temp[16];
    int i, j;

    printf("Please enter ten strings: \n");
    for (i = 0 ; i < 10 ; i++)
        gets(strings[i]);

    for (i = 0 ; i < 9 ; i++)
        for (j = 9 ; j > i ; j--)
            if (strcmp(strings[j], strings[j - 1]) < 0) {
                strcpy(temp, strings[j]);
                strcpy(strings[j], strings[j - 1]);
                strcpy(strings[j - 1], temp);
            }

    for (i = 0 ; i < 10 ; i++)
        printf("%s \n", strings[i]);
}
```

.22

1.

```
/* note: this is not a complete program,
 * because no main() function is defined.
 */
int FindLetters(char str[])
{
    int i, count = 0;

    for (i = 0 ; str[i] != '\0' ; i++)
        if (str[i] >= 'a' && str[i] <= 'z' ||
            str[i] >= 'A' && str[i] <= 'Z')
            count++;

    return count;
}
```

2.

```
/* note: this is not a complete program,
 * because no main() function is defined.
 */
int FindChar(char str[], char ch)
{
    int i, count = 0;

    for (i = 0 ; str[i] != '\0' ; i++)
        if (str[i] == ch)
            count++;

    return count;
}
```

3.

```
/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <string.h>
int FindShortest(char str1[], char str2[],
                 char str3[], char str4[])
{
    int len = strlen(str1);

    if (len > strlen(str2))
        len = strlen(str2);

    if (len > strlen(str3))
        len = strlen(str3);

    if (len > strlen(str4))
        len = strlen(str4);

    return len;
}
```

1.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
int GetLength(char str[])
{
    int len = 0;

    while (str[len] != '\0')
        len++;

    return len;
}

```

2.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
void CopyStrings(char target[], char source[])
{
    int i;

    for (i = 0 ; source[i] != '\0' ; i++)
        target[i] = source[i];

    target[i] = '\0';
}

```

3.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
void ConcatenateStrings(char target[], char source[])
{
    int i, j;

    i = 0;
    while (target[i] != '\0')
        i++;

    for (j = 0 ; source[j] != '\0' ; j++)
        target[i + j] = source[j];

    target[i + j] = '\0';
}

```

4.

```
/* note: this is not a complete program,
 * because no main() function is defined.
 */
int FindLexOrder(char str1[], char str2[])
{
    int i;

    for (i = 0 ; str1[i] == str2[i] ; i++)
        if (str1[i] == '\0')
            return 0;

    return str1[i] - str2[i];
}
```

.24

1.

```
/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <stdio.h>
#include <string.h>
int GetLengths(char strings[][20], int n)
{
    int i, len = 0;

    printf("Please enter strings: \n");
    for (i = 0 ; i < n ; i++) {
        gets(strings[i]);
        len += strlen(strings[i]);
    }

    return len;
}
```

2.

```
/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <string.h>
int GetLexFirst(char strings[][10], int n)
{
    int i, index = 0;
```

```

for (i = 1 ; i < n ; i++)
    if (strcmp(strings[i], strings[index]) < 0)
        index = i;

return index;
}

```

3.

```

/* note: this is not a complete program,
 * because no main() function is defined.
 */
#include <string.h>
void SortByLen(char strings[][20], int n)
{
    int i, j;
    char temp[20];

    for (i = 0 ; i < n - 1 ; i++)
        for (j = n - 1 ; j > i ; j--)
            if (strlen(strings[j]) < strlen(strings[j - 1])) {
                strcpy(temp, strings[j]);
                strcpy(strings[j], strings[j - 1]);
                strcpy(strings[j - 1], temp);
            }
}

```