

תרגיל 3 - מערכים**הגשה עד 25/12/16 בשעה 23:50**

הוראות ההגשה ודגשים מיוחדים מופיעים בסוף התרגיל! **חובה לקרוא ולפעול לפיהן.**

שאלה 1 (10 נקודות)

מילה היא פלינדרום אם כשקוראים אותה מההתחלה לסוף ומהסוף להתחלה מתקבל אותו הדבר. לדוגמא: abcba.

כתוב פונקציה בשם palindroms המקבלת כפרמטר מחרוזת. המחרוזת מחולקת למילים באמצעות רווחים. הפונקציה לא משנה את המחרוזת, ומחזירה את כמות המילים במחרוזת שהן פלינדרום. כתוב את הפונקציה הראשית בתכנית. הפונקציה קולטת מהמשתמש מחרוזת שיכולה להכיל רווחים. לאחר מכן היא קוראת לפונקציה palindrom, ומדפיסה את הערך המוחזר.

אין להשתמש בפונקציות ספריה של מחרוזות.

דוגמא לריצת התכנית:

Enter string: fgfgf cbmkh 8998 mm f6u8 yytyy r5
4

שאלה 2 (10 נקודות)

כתוב פונקציה:

int divide(int a[], int size, int num)

הפונקציה מדפיסה את כל אברי המערך a המתחלקים ב-num ללא שארית, ומחזירה את כמות המספרים שהודפסו. אם num=0 היא מחזירה -1.

כתוב את הפונקציה הראשית בתכנית. היא קוראת פעמיים לפונקציה divide עם מערכים בגדלים 5 ו-10, ועם ערכים של num שהוקלדו על-ידי המשתמש. הפונקציה הראשית מדפיסה את הערכים המוחזרים של divide. אם הערך המוחזר הוא -1 אז מודפסת הודעת שגיאה.

לדוגמא:

Enter 5 numbers: 23 40 3 22 20
Enter num: 10
40 20
2 numbers in the array are divided by 10.
Enter 10 numbers: 99 88 77 66 55 999 888 777 666 555
Enter num: 3
99 66 999 888 777 666 555
7 numbers in the array are divided by 3.

שאלה 3 (80 נקודות)

מטרת תרגיל זה היא סימולציה של המשחק צוללות. חוקי המשחק מוגדרים באתר הבא:
[https://he.wikipedia.org/wiki/%D7%A6%D7%95%D7%9C%D7%9C%D7%95%D7%A \(%D7%9E%D7%A9%D7%97%D7%A7\)A](https://he.wikipedia.org/wiki/%D7%A6%D7%95%D7%9C%D7%9C%D7%95%D7%A (%D7%9E%D7%A9%D7%97%D7%A7)A)

בתרגיל זה המטרה היא לבצע סימולציה בלבד של המשחק. כלומר, יש רק לוח אחד ושחקן אחד. השחקן מציב כלים על הלוח ואח"כ יורה בהם.

המשחק ממומש באמצעות מערך דו-מימדי בגודל 10×10 של שלמים. שימו לב: מבחינת המשתמש הקואורדינטות הן מ-1 עד 10, בעוד שהאינדקסים של המערך הם בין 0 ל-9. כלומר, אם המשתמש בחר באינדקס $[[i],[j]]$, אז צריך לסמן את האיבר $[j-1][i-1]$ במערך.

בכל מקום במערך יכולים להיות הערכים הבאים:

0 – מקום שאין בו צוללת ושלא נפגע.

1 – מקום שאין בו צוללת ושנפגע.

2 – מקום שיש בו צוללת ושלא נפגע.

3 – מקום שיש בו צוללת ושנפגע.

התכנית צריכה לכלול את הפונקציות הבאות.

(5 נקודות) פונקציה `new_game` המאפסת את המערך.

(25 נקודות) פונקציה

`int insert(int board[10][10], int line, int col, int length, int horizon)`

הממקמת צוללת במערך. הפונקציה מקבלת כפרמטרים את לוח המשחק (מערך דו-מימדי), את 2 הקואורדינטות של תחילת הצוללת, את אורכה של הצוללת, והאם הצוללת ממוקמת בצורה אופקית (פרמטר horizon שווה 1) או אנכית (פרמטר horizon שווה 0). הפונקציה בודקת אם מותר להציב את הצוללת במקום המבוקש (אין חריגה מהלוח, הצוללת לא באותו מקום בו נמצאת צוללת אחרת, והצוללת לא סמוכה לצוללות אחרות. גם סמיכות באלכסון אסורה.) אם מותר להציב את הצוללת במקום המבוקש, הפונקציה משנה את המערך בהתאם, ומחזירה 1. אחרת, המערך לא משתנה, והערך המוחזר הוא 0.

לדוגמא, אם `line=3, col=8, horizon=1, length=5`, אז המיקום המבוקש של הצוללת הוא בשורה 3 עמודות 8-12. במקרה כזה הפונקציה תחזיר 0, ולא תשנה את המערך.

(5 נקודות) פונקציה `insert_all`, הממקמת את כל הצוללות באמצעות הפונקציה `insert`. היא מבקשת מהמשתמש להכניס את המיקום והכיוון של צוללת באורך 5, לאחר מכן צוללת באורך 4, שתי צוללות באורך 3, ולבסוף צוללת באורך 2. בכל פעם שהפונקציה `insert` מחזירה 0, המשתמש מקבל הודעת שגיאה, ומתבקש להקליד נתונים חלופיים. דוגמא לריצת הפונקציה:

```
Enter coordinates of 5 length submarine: 2 3
Enter direction of submarine (h/v): h
Enter coordinates of 4 length submarine: 5 2
Enter direction of submarine (h/v): h
Enter coordinates of 3 length submarine: 4 4
Enter direction of submarine (h/v): v
Bad location. Try again.
Enter coordinates of 3 length submarine: 7 7
Enter direction of submarine (h/v): v
Enter coordinates of 3 length submarine: 1 9
Enter direction of submarine (h/v): h
Bad location. Try again.
Enter coordinates of 3 length submarine: 3 8
Enter direction of submarine (h/v): h
Bad location. Try again.
Enter coordinates of 3 length submarine: 1 9
Enter direction of submarine (h/v): v
```

Enter coordinates of 2 length submarine: 7 2

Enter direction of submarine (h/v): h

הניסיון למקם צוללת בקואורדינטות (4,4) נכשל, משום שיש התנגשות עם הצוללת שממוקמת בקואורדינטות (5,2).

הניסיון למקם צוללת בקואורדינטות (1,9) אופקי נכשל, משום שיש חריגה מלוח המשחק. הניסיון למקם צוללת בקואורדינטות (3,8) נכשל, משום שיש קירבה לצוללת אחרת.

(5 נקודות) פונקציה **print** המדפיסה את לוח המשחק, ואת מספר היריות. לדוגמא, אחרי ריצת הפונקציה `insert_all` עם קלט מהמשתמש כמו בדוגמא קודמת, הפלט יהיה:

```
0 0 0 0 0 0 0 0 2 0
0 0 2 2 2 2 2 0 2 0
0 0 0 0 0 0 0 0 2 0
0 0 0 0 0 0 0 0 0 0
0 2 2 2 2 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0 0 0
0 2 2 0 0 0 2 0 0 0
0 0 0 0 0 0 2 0 0 0
0 0 0 0 0 0 0 0 0 0
```

0 trials.

(20 נקודות) פונקציה **shoot** המקבלת כפרמטרים את לוח המשחק וזוג קואורדינטות, מבצעת ירי למיקום זה, ומעדכנת את המערך בהתאם. ערך מוחזר:

1 - קואורדינטות מחוץ ללוח

0 - החטאה

1 - פגיעה

2 - הטבעה

3 - ניצחון

(15 נקודות) פונקציה **play**. תחילה, הפונקציה קוראת ל-`new_game` ול-`insert_all`. לאחר מכן היא מבקשת מהמשתמש שוב ושוב להקליד זוג קואורדינטות. עם כל זוג קואורדינטות שהמשתמש מקליד קוראים לפונקציות `shoot` ו-`print`, ומדפיסים הודעה מתאימה. כך ממשיכים עד שהפונקציה `shoot` מחזירה ערך 3. לאחר ש-`shoot` מחזירה 3, שואלים את המשתמש אם הוא מעוניין במשחק נוסף. אם המשתמש רוצה לשחק פעם נוספת, קוראים ל-`new_game` ול-`insert_all`, וממשיכים בהתאם. אחרת, התוכנית מסתיימת.

לדוגמא, עבור לוח המשחק שהודפס מקודם:

Enter coordinates: 2 5

HIT !!!

```

0000000020
0022322020
0000000020
0000000000
0222200000
0000000000
0000002000
0220002000
0000002000
0000000000

```

1 trials.

Enter coordinates: 13 5
 Out of board !!!
 Enter coordinates: 3 5
 MISS !!!

```

0000000020
0022322020
0000100020
0000000000
0222200000
0000000000
0000002000
0220002000
0000002000
0000000000

```

2 trials.

(5 נקודות) כתוב את הפונקציה הראשית ואת התכנית כולה. מותר להגדיר משתנים נוספים ופונקציות נוספות. השתמש ב-define וב-enum. להשתמש ב-enum עבור ערכי המערך הדו-מימדי ועבור ערכים מוחזרים של shoot. אין להשתמש במשתנים גלובליים.

סעיף בונוס (20 נקודות)

כתוב תכנית לניהול מלאי של מכוניות. לכל מכונית שבמלאי יש 3 מאפיינים:

- יצרן: אחד מהאפשרויות Toyota, Ford, Fiat, Nissan
- שנת יצור: מספר שלם בין 2014 ל-2016.
- צבע: אחד מהאפשרויות: לבן, שחור, צהוב, כחול, אדום.

מגדירים בתכנית:

```

enum model { Toyota, Ford, Fiat, Nissan };
enum color { white, black, yellow, blue, red, green };

```

התוכנית מכילה מערך תלת מימדי $a[4][3][6]$, כאשר אינדקס ראשון מתאר את היצרן, אינדקס שני מתאר את שנת הייצור ואינדקס שלישי מתאר את הצבע. האיבר $a[i][j][k]$ מכיל את כמות הרכבים שבמלאי שיש להם את היצרן, שנת יצור וצבע מבוקשים. לדוגמא, אם $a[0][0][0]=4$ אז יש במלאי 4 רכבים מדגם טויוטה משנת 2014 בצבע לבן. דוגמא נוספת: אם $a[1][2][3]=0$ אז אין במלאי רכבים מדגם פורד משנת 2016 בצבע כחול.

כתוב את הפונקציות הבאות:

void add(int a[4][3][6], int model, int year, int color)

מוסיפה רכב אחד למאגר.

int find(int a[4][3][6], int model, int year, int color)

מחזירה את כמות הרכבים במלאי עם דגם, שנת יצור וצבע נתונים.

int find_model(int a[4][3][6], int model)

מחזירה את כמות הרכבים מדגם מסויים.

void find_year(int a[4][3][6], int year)

מחזירה את כמות הרכבים משנת ייצור מסוימת.

int frequen_color(int a[4][3][6])

מחזירה צבע שממנו יש הכי הרבה רכבים במאגר. (אם יש מספר צבעים כאלו, היא מחזירה אחד מהם.)

כתוב את הפונקציה הראשית בתוכנית. הפונקציה מאפסת את המערך התלת מימדי, וקוראת לפונקציות האחרות לפי בחירה של המשתמש מתפריט.

הערות:

1. אין להשתמש בתרגיל, למעט סעיף בונס, בחומר שטרם נלמד, או שנלמד לאחר נושא התרגיל.

2. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום שצריך להכניס מספר שלם – נכניס מספר שלם (ולא שבר או אות). אנחנו לא מתחייבים שהוא יהיה חיובי או א-שלילי, או בטווח מסוים! – אלא אם כן נאמר אחרת בשאלה עצמה.

3. בכל פעם שהמשתמש מקליד קלט שגוי התוכנית מבקשת קלט חוזר.

4. אחרי כל הדפסה יש לבצע ירידת שורה.

5. בתרגיל יש להשתמש בספריות stdio בלבד.

6. יש להקפיד על תכנות נכון:

a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים

כ: **enum** או **const**, בהתאם לצורך.

b. יש לרשום הערות.

c. יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים!

d. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן.

e. לפני בקשת קלט (scanf) יש להדפיס למשתמש הוראה (printf) איזה קלט מבוקש.

f. יש להקפיד על מוסכמות התכנות הנכון (שמות כמו שצריך וכו').

g. יש להקפיד על כל כללי התכנות הנכון כפי שנלמדו בכיתה.

7. בהצלחה ☺

הנחיות הגשה

- תרגילים הם ביחידים! כל עבודה משותפת היא אסורה ותיענש בחומרה!
- כל התרגילים צריכים להיות בתוך קובץ אחד הכולל אותם לפי הסדר.
- ההגשה היא של קבצי הקוד (קובצי .c) בלבד, שישלחו אך ורק דרך המערכת! יש לכווץ את כל הקבצים .c לקובץ אחד בפורמט ZIP או RAR להגיש אותו.
- שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראי, דוד, במייל: davidt@sce.ac.il
- בתחילת הקובץ יש להוסיף את התיעוד הבא:

/* Assignment: 2

Campus: Ashdod / Beer Sheva (תבחרו את המתאים)

Author: Israel Israeli, ID: 01234567

*/

- כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.
- הארכות יינתנו אך ורק במקרים חריגים (מילואים, אבל על קרובים ומחלה חריפה!) ובצרוף אישורים מתאימים. כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה האחראית על התרגיל לפחות יומיים לפני חלוף הדד-ליין!
- ההגשה היא עד התאריך האחרון לתרגיל: 25/12/16 בשעה **23:50**. הגשה מאוחרת אפילו בדקה – לא תתקבל (המערכת חוסמת את אפשרויות ההגשה!). קחו זאת בחשבון ותכננו את זמנכם בהתאם!