

תרגיל 1 – פונקציות מערכים ומחרוזות

הוראות ההגשה ודגשים מיוחדים מופיעים בסוף התרגיל! **חובה לקרוא ולעקוב אחריהן.**

בתרגיל זה עליכם לממש את כל מה שכתוב בתרגיל.

בתרגיל יש להשתמש בספריות stdio בלבד.

על הפתרון הסופי להופיע ללא main, חלק זה יתווסף על-ידנו בעת הבדיקה (כמובן יש לבדוק את הפתרון בהרצה לפני ההגשה)

הקדמה:

מתכנת צעיר המתקבל לעבודה בחברה "ראש טוב", בתור המשימה הראשונה שלו בקריירה הוא התבקש לבנות מספר פונקציות עבור פרויקט מסוים. את הדרישות וההסברים על הפונקציות תוכלו לראות בהמשך.

הנקודות החשובות:

- ✓ המתכנת חייב לשמור על כל החתימות של הפונקציות (כולל השמות המדויקים, ערך החזרה ורשימת הפרמטרים).
- ✓ על הפתרון הסופי להופיע ללא main, חלק זה יתווסף על-ידנו בעת הבדיקה (כמובן יש לבדוק בהרצה לפי ההגשה)

הפונקציות הנדרשות :

(1) **היסטוגרמה** היא צורת הצגה גרפית של נתונים. נרצה להציג את מספר ההופעות של איברי המערך הנתון בעזרת ההיסטוגרמה.

כתוב פונקציה בשם **histogramArray** המקבלת מערך מספרים שלמים בטווח בין 0 ל-20 ואת גודל המערך ומדפיסה היסטוגרמה של המספרים, ז"א לכל ערך המופיע במערך הפונקציה מדפיסה את מספר הכוכביות המייצגות את כמות ההופעות של המספר במערך הנתון. **מותר לעבור על המערך הנתון פעם אחת בלבד.**

למשל, הפונקציה מקבלת את המערך הבא ואת גודל מערך - 14 :

13, 1, 3, 4, 7, 3, 13, 9, 13, 5, 4, 1, 4, 13

הפונקציה תדפיס את ההיסטוגרמה הבאה:

Value	Histogram
1	**

```

3      **
4      ***
5      *
7      *
9      *
13     ****

```

(2) כתוב פונקציה בשם **printMaxPartAscending** המקבלת מערך מספרים שלמים ואת גודל המערך ומדפיסה את החלק הארוך ביותר בו איברי הקטע מסודרים בסדר עולה, במידה ולא קיים קטע העונה על דרישה זו, תודפס הודעה (ראה בדוגמה למטה) לדוגמה, הפונקציה מקבלת את המערך הבא וגודל מערך – 10:

-2, 8, 23, -160, -74, 8, 85, -47, 12, 7

הפונקציה מדפיסה :

-160 -74 8 85

הפונקציה מקבלת את המערך הבא וגודל מערך – 10:

10,9,8,7,6,5,4,3,2,1

הפונקציה מדפיסה : "No such segment!"

(3) איבר במטריצה יקרא "אוכף" אם הוא הכי גדול בעמודתו והכי קטן בשורתו . כמובן שאיבר כזה אינו תמיד קיים .

כתוב פונקציה **checkSaddleElement** המקבלת מטריצה (כאשר מספר העמודות 10), מספר המהווה את מספר השורות ומספר המהווה את מספר העמודות בהתאמה ומחזירה 1 (אמת) אם המטריצה מכילה איבר "אוכף" אחרת הפונקציה מחזירה 0 (שקר)

למשל הפונקציה מקבלת את המטריצה הבאה, מספר שורות 8 , ומספר עמודות 10:

```

1  2  0  3  9  4  5  3  6  7
2  4  3  5  9  0  2  3  4  1
5  3  4  7  6  1  9  0  4  2
6  9  7  8  6  7  7  4  5  6
8  3  1  9  2  0  6  2  8  2
2  7  3  0  3  6  3  1  5  3
8  2  5  9  7  7  8  3  7  3

```

1 7 4 8 5 8 5 0 0 6

ומחזירה 1.

(4) כתוב פונקציה בשם **rotateMatrix90** המקבלת מערך דו-מימדי (מטריצה) ריבועית (כאשר מספר העמודות 3), מספר המהווה את מספר השורות ומספר המהווה את מספר העמודות בהתאמה ומסובבת את המטריצה בסיבוב של 90, מעלות בכיוון השעון .

למשל פונקציה מקבלת את המטריצה הבאה, מספר שורות 3 ומספר עמודות 3:

1	2	3
4	5	6
7	8	9

מטריצה המסובבת:

7	4	1
8	5	2
9	6	3

(5) כתוב פונקציה בשם **rotateString** המקבלת מחרוזת, מספר שלם המהווה את מספר התווים שיש להזיז ומספר נוסף המהווה את כיוון ההזזה - שמאלה או ימינה 1 עבור הזזה שמאלה, 2 עבור הזזה ימינה ומעדכנת את המחרוזת תוך כדי הזזה, למשל :

עבור קריאה: (**rotateString** ("refdeae",2,1) הפונקציה מזיזה 2 תווים שמאלה ומעדכנת את המחרוזת ל: "fdeeaere".

עבור קריאה: (**rotateString** ("refdeae",10,1) הפונקציה מזיזה 10 תווים שמאלה ומעדכנת את המחרוזת ל: "fdeeaere".

עבור קריאה: (**rotateString** ("refdeae",2,2) הפונקציה מזיזה 2 תווים ימינה ומעדכנת את המחרוזת ל: "aerefdee".

הערות :

- אסור להשתמש בספריית string.h

- יש לבצע באמצעות מספר ההזזות המינימלי.

(6) כתוב פונקציה בשם **countWords** המקבלת מטריצה המכילה את קודי ה-ASCII (כאשר מספר העמודות 8), מספר המהווה את מספר השורות ומספר המהווה את מספר העמודות בהתאמה ומחרוזת המכילה מילה. הפונקציה תספור ותחזיר כמה פעמים המילה מהמחרוזת מופיעה במטריצה.

כיווני קריאה : משמאל לימין , מלמלה למטה ובאלכסון .
יש להשתמש בפונקציות עזר.

לדוגמה הפונקציה מקבלת מטריצה הבאה , מספר שורות 6, ומספר עמודות 8

```
'a'  'w'  'o'  'r'  'd'  'b'  'c'  'w'
'd'  'o'  'o'  'e'  'f'  'h'  'i'  'o'
'j'  'r'  'k'  'r'  'w'  'l'  'w'  'r'
'm'  'd'  'n'  'o'  'd'  'o'  'o'  'd'
'p'  'q'  'r'  's'  'w'  'o'  'r'  'd'
't'  'u'  'v'  'w'  'x'  'y'  'd'  'd'
```

ומחרוזת המכילה מילה "word". הפונקציה תחזיר מספר – 7.

(7)

הפונקציה **decipher** - מקבלת רצף מילים מוצפן (**ciphered token**)

(**sequence**) ממירה ומדפיסה אותו לרצף מילים מפוענח (קריא)

○ אופן ההצפנה: התו הראשון יומר לתו חדש בעל קוד ASCII הקטן באחד, התו השני יומר לתו חדש בעל קוד ASCII הקטן בשניים וכך הלאה....

○ דוגמא למשפט מוצפן –

"Btwlzx Dqqes Eq|pj4 Tjlvqujs lqoqjy bpg Eqfxtx Xcwwtt5"

○ המשפט הקריא לאחר התרגום –

○ "Arthur Conan Doyle. Sherlock Holmes and Doctor Watson."

○ הטבלה הבאה ממחישה את צורת התרגום של המילה הראשונה (Btwlzx):

קריא	מוצפן	פער
A	B	1
r	t	2
t	w	3
h	l	4
u	z	5
r	x	7

8) הפונקציה **FindSimilarWords** - מחפשת "מילים דומות" ומדפיסה אותם.

- a "מילים דומות" – מילים אשר מכילות את אותן האותיות (ללא הבדל בין אותיות גדולות וקטנות), כאשר אין חשיבות לסדר הופעת האותיות או למספר ההופעות של כל אות.
- b למשל Abc ו-Bbaacc הינן מילים דומות (המילים מכילות את אותן האותיות, ללא הבדל בין אותיות קטנות וגדולות וללא הבדל בסדר ובכמות האותיות).
- c Abg ו-Bbaacc אינן מילים דומות (המילים לא מכילות את אותן האותיות)

דוגמת הרצה:

```
decipher("Btwlzx Dqqes Eq|pj4 Tjhvqujs Iqoqjy bpg Eqfxtx Xcwwtt5") //  
output lines 1 findSimilarWords ("Nanny have you any cheap peach ?");  
// output lines 2-3
```

פלט עבור הדוגמה:

```
Arthur Conan Doyle. Sherlock Holmes and Doctor Watson // output lines 1  
Nanny any //output lines 2  
cheap peach // output lines 3
```

הערות:

- הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום שצריך להכניס מספר שלם – נכניס מספר שלם (ולא שבר או תו). אנחנו לא מתחייבים שהוא יהיה חיובי או א-שלילי, או בטווח מסוים – אלא אם כן נאמר אחרת בשאלה עצמה.
- אחרי כל הדפסה יש לבצע ירידת שורה.
- בתרגיל יש להשתמש בספריות stdio בלבד**
- יש להקפיד על תכנות נכון:
 - כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), **חייבים** להיות מוגדרים כ: const, enum, define בהתאם לצורך.
 - יש לרשום הערות עבור כל חלק משמעותי או מורכב ובאנגלית בלבד. יש לתעד כל פונקציה (מהו כל פרמטר, מה תפקידו ומה הערכים המוחזרים)
 - יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים!
 - יש לנסות ולייעל את הקוד והתוכנית ככל שניתן (לא להוסיף פקודות או לולאות מיותרות, לא לרוץ בלולאה מעבר למה שצריך, וכו')

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

f. יש להפקיד לבצע חלוקה לפונקציות. אין לכתוב קוד כפול. נסו לחשוב איזה חלוקה תיתן לכם קוד קריא וקל יותר לתחזוק ושימוש מחודש.

g. יש להקפיד על כל מוסכמות וכללי התכנות הנכון כפי שנלמדו בכיתה.

בהצלחה

הנחיות הגשה:

- הגשת תרגילים בזוגות בלבד.
- הגשה היא בקובץ C בלבד תחת השם assignment1.c, הגשה תחת שם אחר תקבל את הציון 0.
- בתחילת הקובץ יש להוסיף את התיעוד הבא עם השמות והמספרים שלכם:

```
/* Assignment: 1
```

```
Author1: Israel Israeli, ID: 01234567
```

```
Author2: Canda Canadi, ID: 76543210
```

```
*/
```