

תרגיל 3 – פונקציות מערכים ומחרוזות**הגשה עד 26/12/19****הוראות הגשה:**

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק לגב' סבטלנה רוסין, במייל: sceassign2016@gmail.com.
 2. תרגילים הם ביחידים! כל עבודה משותפת היא אסורה ותיענש בחומרה!
 3. להגיש רק תכניות שעוברות קומפילציה על מהדר שפת C gcc Ubuntu Linux.
 4. ההגשה היא של קבצי הקוד (כקובץ c). בלבד. יש ליצור 3 קבצים: part1.c, part2.c ו- part3.c, לכוון את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
 5. בתחילת הקובץ יש להוסיף את התיעוד הבא:
- ```

/* Assignment: 3
 Author: Israel Israeli, ID: 01234567
*/

```
- כמובן שיש לעדכן את השמות ומספרי תעודות הזוהות שלכם.
6. 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!
  7. ההגשה היא עד התאריך האחרון לתרגיל: 26/12/19 בשעה 23:50. הגשה מאוחרת אפילו בדקה – לא תתקבל (המערכת חוסמת את אפשרויות ההגשה!). קחו זאת בחשבון ותכננו את זמנכם בהתאם!
  8. מותר לכם ומומלץ ליצור פונקציות עזר.
  9. אין להשתמש בתרגיל בחומר שטרם נלמד, או שנלמד לאחר נושא התרגיל, אלא אם נכתב במפורש בתרגיל שמותר.
  10. 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.
  11. בכל פעם שהשתמש מקליד קלט שגוי התוכנית מבקשת קלט חוזר.
  12. אחרי כל הדפסה יש לבצע ירידת שורה.
  13. בתרגיל יש להשתמש בספריית stdio, בלבד!
  14. יש להקפיד על תכנות נכון:
- a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: const, define או enum, 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.
- בהצלחה ☺

### חלק א – מערכים חד-מימדיים – מענה בקובץ part1.c

1. ( 5 נק') כתבו פונקציה בשם **isBalanced** המקבלת מערך של מספרים שלמים, כולל את גודלו ובודקת האם קיים אינדקס אמצעי במערך ומחזירה אותו, אחרת הפונקציה מחזירה ערך -1.
- אינדקס אמצעי הוא תא אשר סכום כל האיברים עד אליו וכולל, שווה לסכום שאר האיברים במערך הנמצאים הלאה ממנו. לדוגמא,
- הפונקציה מקבלת את המערך : {1,4,6,2,10,3} קיים אינדקס אמצעי, והוא 3 (התא שבו נמצא המספר 2), לכן הפונקציה מחזירה 3 .
- הפונקציה מקבלת את המערך : {1,4,6,12,10,3} לא קיים אינדקס אמצעי, לכן הפונקציה מחזירה -1 .
2. ( 15 נק') כתבו פונקציה בשם **CommonPart** המקבלת שני מערכים של מספרים שלמים, כולל את גודלם ומדפיסה את החלק החופף (המשותף בין סופו של המערך הראשון וההתחלה של המערך השני) באורך המקסימלי ואת אורכו, לדוגמא :
- ```
int first []={ 0,1,2,3,4,1,3,2,1,4,1,3,2,1 };
int second[]={1,3,2,1,4,1,3,2,1,3,2,1 };
```
- החלקים החופפים הם:
- 0,1,2,3,4,1,3,2,1,4,1,3,2,1
- 1,3,2,1,4,1,3,2,1,3,2,1
- 0,1,2,3,4,1,3,2,1,4,1,3,2,1
- 1,3,2,1,4,1,3,2,1,3,2,1

0,1,2,3,4,1,3,2,1,4,1,3,2,1

1,3,2,1,4,1,3,2,1,3,2,1

הפונקציה מדפיסה :

Max common part : { 1,3,2,1,4,1,3,2,1 } , Length = 9

במידה ולא קיים חלק כזה תודפס הודעה מתאימה.

(2 נק') כתבו תוכנית ראשית שתיצור 3 מערכים : ראשון בגודל 6 , שני בגודל 14 ושלישי בגודל 12 . התוכנית תמלא את כל המערכים בקלט מהמשתמש ותשלח את המערך הראשון לפונקציה *isBalanced* ותדפיס את "האינדקס אמצעי" במידה והוא קיים, או הודעה מתאימה במידה והוא לא קיים, לאחר מכן התוכנית תשלח את 2 המערכים הנוספים לפונקציה *CommonPart* .

הערות:

ניתן ליצור מערכים בגודל קטן יותר מהמוגדר במשימה

יש להגדיר את גדלי המערכים בעזרת *define*.**חלק ב – מערכים דו-מימדיים – מענה בקובץ part2.c**3. (10 נק') כתבו פונקציה *MultipliersDividers* המקבלת מטריצה ריבועית (מספר שורות

שווה למספר עמודות) ושני מספרים נוספים המהווים מיקום (אינדקס שורה ואינדקס עמודה) של

איבר במערך ומדפיסה את כמות המספרים במערך שהם כפולות של האיבר הזה וכמו כן את

כמות המספרים שהוא מכפלה שלהם. יש לבדוק שהמיקום הוא תקין לפי מימדי המטריצה.

דוגמא 1:

הפונקציה מקבלת מטריצה (בדוגמה הוא בגודל 4*4)

250	90	1100	45
5	10	400	81
21	34	55	67
50	67	3	245

ושני מספרים: 0 ו-4 , בדוגמא זו ניתן לראות כי הנתונים אינם חוקיים (אינדקס עמודה במקרה זה אינו קיים במערך הנתון) ולכן הפונקציה תדפיס:

Invalid column index!

דוגמא 2:

הפונקציה מקבלת מטריצה (בדוגמה הוא בגודל 4*4)

250	90	1100	45
5	10	400	81
21	34	55	67
50	67	3	245

ושני מספרים: 3 ו-0 (הערך 50) ולכן התכנית תדפיס:

Number of multiples : 3 (1100,250,400)

Number of divisors : 2 (5,10)

*שימו לב שאת הסוגריים (מופיעים ב-Italics) אין צורך להדפיס.

4. (20 נק') כתבו פונקציה בשם **MaxZeroArea** המקבלת מטריצה בגודל M שורות ו-N עמודות

שמכילה רק 0 או 1 ומספר נוסף length. הפונקציה תמצא את תת-המטריצה הריבועית בגודל length * length שמכילה הכי הרבה אפסים.

הפונקציה תדפיס את פרטי התת-מטריצה לפי הדוגמה ותחזיר את כמות האפסים המקסימלית. יש לבדוק האם ערך length הוא חוקי לפי המימד הקטן ביותר של המטריצה (בדוגמה המופיעה בהמשך לפי כמות השורות), במידה וlength לא חוקי הפונקציה תדפיס את הודעה רלוונטית ותחזיר 0. במידה וקיימים מספר קטעים של תת מטריצה העונים לדרישה, יש להדפיס את פרטי האחד מהם. למשל, הפונקציה מקבלת את המטריצה הבאה (8 שורות ו-10 עמודות):

0	1	1	1	0	1	0	0	1	0
0	1	0	1	1	0	0	0	0	0
1	0	0	0	0	0	1	0	0	1
0	1	0	1	1	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	1	0	1	0	1	0	1	0	1
0	0	0	1	1	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0

4 length -

הפלט :

0	0	0	0
0	1	0	0
1	0	0	0
0	0	0	0

Sub matrix starts at indexes [1][5] , length = 4 : 14 Zeroes

(2 נק') כתבו תוכנית ראשית שתיצור 2 מערכים דו-מימדיים : הראשון בגודל 4 * 4 והשני בגודל 8 שורות ו-10 עמודות (אבל אפשר גם ליצור מערך דו-מימדי קטן יותר), יש לאתחל את המערכים בקלט מהמשתמש (למערך השני יש לאפשר להכניס רק 0 או 1, אחרת יש לדרוש הזנה חוזרת). יש לקלוט גם ערכים נוספים כמו אורך הקטע (length) וכו. לאחר מכן התוכנית תפעיל את הפונקציות MultipliersDividers ו-MaxZeroArea.

שימו לב, במידה והפונקציה מחזירה ערך, יש להדפיס אותו, במידה ותת מטריצה לא קיימת יש להדפיס את ההודעה : Length is illegal.

חלק ג – מחרוזות – מענה בקובץ part3.c

5. (5 נק') כתבו פונקציה בשם **ReverseLetters** המקבלת כפרמטר מחרוזת. הפונקציה הופכת את סדר האותיות בכל מילה. תו רווח הוא התו המפריד היחיד בין מילים.

לדוגמא,
המחרוזת:

“one two three four five”

תשונה ל:

“eno owt eerht ruof evif”

6. (19 נק') כתבו פונקציה בשם **ReverseWords** המקבלת כפרמטר מחרוזת. הפונקציה הופכת את סדר המילים במחרוזת. תו רווח הוא התו המפריד היחיד בין מילים. לדוגמא,

המחרוזת:

“one two three four five”

תשונה ל:

“five four three two one”

7. (5 נק') כתבו פונקציה בשם **UpperLower** המקבלת כפרמטר מחרוזת. הפונקציה הופכת כל אות קטנה לגדולה, וכל אות גדולה לקטנה. שאר התווים ללא שינוי.

לדוגמא,

המחרוזת:

“one TWO three 4five6+sEvEn”

תשונה ל:

“ONE two THREE 4FIVE6+SeVeN”

8. אנגרמה – אַנגְרָמָה היא יצירת מילה חדשה מערבוב אותיותיה של מילה קיימת, או משפט חדש מערבוב אותיות של משפט קיים. נקראת לעיתים "חושחש" (15 נק') כתבו פונקציה בשם **FindAnagrams** המקבלת מערך מחרוזות בגודל 8 (גודל כל שורה 50 תווים לכל היותר). הפונקציה תדפיס את כל המילים שהן אנגרמות (מכילות אותיות זהות). במידה ולא קיימות אנגרמות, הפונקציה תדפיס הודעה מתאימה. אין צורך להבדיל בין אות קטנה וגדולה. למשל,

Spare

Abcd

xyz

pears

Restful

Car

Motor

fluster

הפונקציה תדפיס

Spare pears

Restful fluster

(2 נק') כתבו את הפונקציה הראשית. הפונקציה יוצרת מחרוזות ומערך מחרוזות, ממלא אותם בנתונים כקלט מהמשתמש. לאחר מכן היא קוראת לכל אחת מ-3 הפונקציות הראשונות עם מחרוזת הקלט, ומדפיסה את המחרוזת לאחר השינוי. מותר להשתמש במחרוזת עזר.

הערות:

- שלושת הפונקציות הראשונות לא מחזירות ערך ולא מדפיסות דבר. הן משנות את המחרוזת שקיבלו כפרמטר.
- ניתן להניח כי האורך של מחרוזת אינו עולה על 50 תווים.
- מחרוזת הקלט יכולה להכיל רווחים בודדים וכן רצפים של רווחים.

לאחר מכן פונקציה הראשית מפעילה את הפונקציה **FindAnagrams**.

בהצלחה !!