

תרגיל 3 – פונקציות מערכים ומחרוזות**הגשה עד 13/12/18****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, לכווץ את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
 5. בתחילת הקובץ יש להוסיף את התיעוד הבא:
- ```

/* Assignment: 3
 Campus: Ashdod / Beer Sheva (תבחרו את המתאים)
 Author: Israel Israeli, ID: 01234567
*/

```
- כמובן שיש לעדכן את השמות ומספרי תעודות הזרות 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: 13/12/18 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: `define`, `const` 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') כתוב פונקציה `int PrintDivided ( int arr[],int size, int value)` המקבלת מערך של מספרים שלמים `arr` וגודלו `size` וערך נוסף `value`. הפונקציה מדפיסה את כל אברי המערך המתחלקים ב-`value` ללא שארית ומחזירה את כמות המספרים שהודפסו, אם `value=0` היא מחזירה -1.

למשל,

המערך `arr`: 20 22 3 40 23, הערך `value`: 10

הפונקציה מדפיסה: 20 40 ומחזירה 2.

המערך `arr`: 20 22 3 40 23, הערך `value`: 0

הפונקציה לא מדפיסה שום דבר ומחזירה -1.

2. מערך נקרא פלינדרום אם סדר המספרים שבו אינו משתנה בין קריאה מימין לשמאל ובין קריאה משמאל לימין. לדוגמא, מערך 1 3 3 1 הוא פלינדרום.

(15 נק') כתוב פונקציה `void PrintMaxPalindrom( int arr[],int size)` המקבלת

מערך מספרים שלמים כולל את גודלו ומדפיסה את הרצף הארוך ביותר המקיים את תנאי הפלינדרום באורך של שני איברים לפחות, למשל,

הפונקציה מקבלת:

1 2 1 2 3 7 4 7 3 8

הפונקציה מדפיסה:

3 7 4 7 3

הפונקציה מקבלת:

1 2 6 9 3 5 4 7 3 8

הפונקציה מדפיסה:

No palindrom exist

(4 נק') כתוב תוכנית ראשית שתקלוט ערך שלם חיובי – value עבור הפונקציה הראשונה ותיצור שני מערכים: מערך one שגודלו N, אשר שווה ל-14, ומערך two שגודלו M, אשר שווה ל-10. יש למלא את שני המערכים בקלט מהמשתמש ולהפעיל את כל הפונקציות הנ"ל. שים לב, במידה והפונקציה מחזירה ערך, יש להדפיס אותו. יש להגדיר את גדלי המערכים בעזרת define.

## חלק ב – מערכים דו-מימדיים – מענה בקובץ part2.c

נגדיר טבעת במטריצה (מערך דו-מימדי) באופן הבא:

עבור מטריצה של שלמים בגודל NxN.

נחלק את המטריצה לטבעות.

טבעת 0 היא הטבעת החיצונית של מטריצה.

טבעת 1 היא טבעת פנימית יותר של המטריצה והיא מורכבת מהאיברים בשורה השנייה ובעמודה השנייה מההתחלה ומהסוף, ללא איברים ששייכים לשורה או לעמודה הראשונה והאחרונה.

סדר הטבעות ממשיך הלאה, עד לטבעת האחרונה שהיא הפנימית ביותר.

דוגמה עבור מטריצה 5x5.

|   |   |    |   |   |
|---|---|----|---|---|
| 1 | 2 | 3  | 4 | 5 |
| 1 | 5 | 5  | 5 | 2 |
| 1 | 4 | 30 | 4 | 2 |
| 1 | 2 | 2  | 2 | 2 |
| 1 | 1 | 1  | 1 | 2 |

למטריצה זו שלוש טבעות 0,1,2: 0 – הטבעת החיצונית (הכהה ביותר), 1 – טבעת פנימית יותר, 2 – הטבעת הפנימית ביותר בדוגמה הנ"ל, אשר במקרה הנ"ל מורכבת מהאיבר האמצעי של המטריצה.

דוגמה עבור מטריצה 4x4.

|   |   |    |   |
|---|---|----|---|
| 1 | 2 | 3  | 4 |
| 1 | 5 | 5  | 5 |
| 1 | 4 | 30 | 4 |
| 1 | 2 | 2  | 2 |

3. (18 נק') כתוב פונקציה `int Ring(int matrix[][N], int id)` המקבלת מערך דו-ממדי (מטריצה) של שלמים בגודל N x N ומקבלת מספר שלם id, שמייצג את מספר הטבעת המבוקשת ומחזירה את סכום איברי המטריצה הנמצאים בטבעת זו. במידה ומספר הטבעת אינו חוקי, כלומר מס הטבעת המבוקשת גדול מכמות הטבעות של מטריצה בגודל N x N, או שלילי פונקציה מחזירה 0.

דוגמה 1      עבור מטריצה שבגודל 5 שבדוגמה הקודמת:

id: 1

## פונקציה מחזירה 29

## דוגמה 2

id: 3

## פונקציה מחזירה 0

4. (14 נק') כתב פונקציה בשם `int AllRings(int matrix[][N])` שתחשב ותדפיס את סכום האברים בטבעות המטריצה כפי שהוגדרו. הפונקציה תחזיר את הסכום הכללי של כל הטבעות שבמטריצה.

דוגמה עבור מטריצה  $5 \times 5$  הפונקציה תדפיס :

Ring 0 : 30 ( 1+1+1+1+1+1+1+2+2+2+2+5+4+3+2+1 )

Ring 1: 29      ( 4+2+2+2+4+5+5+5 )

Ring 2: 30 (30)

## הפונקציה תחזיר : 89

(4 נק') כתוב תוכנית ראשית שתיצור מערך דו-מימדי בגודל  $N \times N$  (למשל, 5 שורות ו-5 עמודות), תמלא אותו בערכים כקלט מהמשתמש ותפעיל את כל הפונקציות הנ"ל (יש לקלוט גם ערך עבור מספר הטבעות עבור הפונקציה Ring).

שים לב, במידה והפונקציה מחזירה ערך, יש להדפיס אותו, במידה והטבעת לא קיימת הפונקציה תדפיס : Number is illegal

### חלק ג - מחרוזות- מענה בקובץ part3.c

5. (18 נק') **מחרוזת סוגריים** היא תקינה אם מתקיימים 3 התנאים הבאים:

## 1. כל התווים הם '(' ו-')'

2. מספר ההופעות של ' (') שווה למספר ההופעות של ' (').

3. כל סוג של סוגריים אמור להופיע במחרוזת בצורה לוגית, כלומר לסגור סוגר אחר – ראה דוגמה 2 למקרה בו מתקיימים תנאי 1 ו-2 אך תנאי 3 מופר.

כתוב פונקציה בשם **StringChecker** המקבלת מחרוזת תווים ובודקת האם המחרוזת היא מחרוזת סוגריים תקינה, במידה וכן היא מחזירה אמת, אחרת שקר.

דוגמה 1:

הפונקציה מקבלת :

$$((\ ))((\ ))((\ ))$$

## מחזירה אמת

דוגמה 2:

הפונקציה מקבלת :

(( ))

מחזירה שקר

דוגמה 3:

הפונקציה מקבלת :

((a))

מחזירה שקר

6. (18 נק') הפונקציה בשם **FindSimilarWords** - מקבלת מחרוזת שמורכבת ממילים שמורפרדות ע"י רווח בודד (אין צורך לבדוק את תקינות הקלט), מחפשת "מילים דומות" ומדפיסה אותם.

"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. למשל,

Abc ו-Bbaacc הינן מילים דומות (המילים מכילות את אותן האותיות, ללא הבדל בין אותיות קטנות וגדולות וללא הבדל בסדר ובכמות האותיות).  
Abg ו-Bbaacc אינן מילים דומות (המילים לא מכילות את אותן האותיות).

למשל דוגמה 1:

FindSimilarWords ("Nanny have you any cheap peach ?")

הפונקציה תדפיס:

Nanny any  
cheap peach

דוגמה 2:

FindSimilarWords ("Nannxy have you any cheap porch ?")

הפונקציה תדפיס:

There are no similar words.

הערה: ניתן להניח שאורכה של כל מילה במחרוזת 80 הוא תווים לכל היותר.

( 4 נק') כתוב תוכנית ראשית שקולטת מחרוזת ובודקת האם המחרוזת היא מחרוזת סוגריים תקינה, במידה וכן התוכנית תדפיס "Valid string brackets" אחרת היא תדפיס "Invalid string brackets"

לאחר מכן, התוכנית קולטת מחרוזת נוספת ומדפיסה את כל המילים הדומות המופיעות בה, במידה והמחרוזות אינן מכילות מילים דומות, התוכנית תדפיס הודעה מתאימה.

**בהצלחה !!**