

תרגיל 4 – מצבעים והקצאה דינמית**הגשה עד 14/01/2025 בשעה 23:50****הוראות הגשה:**

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה, גב' 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 Visual Studio C.
4. ההגשה היא של קבצי הקוד (כקובץ c). בלבד. יש ליצור 4 קבצים: `part1.c`, `part2.c`, `part3`-`part4.c`, לכווץ את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהגיש רק קובץ זה.
5. בתחילת הקובץ יש להוסיף את התיעוד הבא:
/* Assignment: 4
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 הدد-ליין!
7. ההגשה היא עד התאריך האחרון לתרגיל 14/01/2025 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 בספריות `malloc`, `stdlib`, `string`, `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: מחרוזת דינאמית (10 נקודות) מענה בקובץ part1.c

1. (8 נק') כתוב פונקציה `NewString(char* str)` המקבלת כפרמטר מחרוזת המורכבת ממספרים, אותיות קטנות ורווחים. ניתן להניח שהמספרים חיוביים ומורכבים מסיפרה אחת או יותר. הפונקציה יוצרת ומחזירה את המחרוזת החדשה בצורה הבאה:
- ספרה 2 תוחלף ב22, הספרה 3 תוחלף ב333, הספרה 4 תוחלף ב4444 וכו'. הספרה 0 לא תופיע כלל, בין מספר למספר יופיעו סימן מינוס (-) לכל רצף של רווחים, וכוכבית (*) לכל רצף של אותיות הקטנות.
- לדוגמה,
- עבור מחרוזת הפרמטר:
- ```
1023 abc 425 hh 231qqq 42
```
- הפונקציה תיצור ותחזיר מחרוזת הבאה:
- ```
122333-44442255555-223331-444422
```

- (2 נק') כתוב את הפונקציה הראשית. הפונקציה הראשית מאתחלת באמצעות המשתמש מחרוזת באורך 50 תווים לכל היותר. היא קוראת לפונקציה הנ"ל ומדפיסה את המחרוזת החדשה. בסוף, התוכנית משחררת את כל הזיכרון הדינאמי.

חלק ב': הקצאה דינמית חד-מימדית (10 נקודות) מענה בקובץ part2.c

- (8 נק') כתוב פונקציה:
- ```
int* SymmetricDiff(int * one, int sizeOne, int * two, int sizeTwo, int *pNewSize)
```
- הפונקציה מקבלת כפרמטרים: שני מערכים ואת גודליהם, ועוד מצביע למספר שלם `pNewSize`, המהווה מצביע לגודל של המערך החדש. הפונקציה יוצרת מערך חדש שיכיל את ההפרש הסימטרי של המערכים. כלומר, את כל האיברים שמופיעים במערך `one` אך לא מופיעים במערך `two` ואת כל האיברים שמופיעים במערך `two` ולא מופיעים במערך `one`. לדוגמה אם הפרמטרים שהתקבלו הם:

one : {1, 3, 4, 6, 10, 13, 14, 7, 8, 9}      sizeOne = 10  
 two : {2, 3, 6, 15, 16, 18, 20, 9, 11, 13}      sizeTwo = 10

אזי הפונקציה תייצר ותחזיר מצביע למערך החדש:

{1,4,10,14,7,8,2,15,16,18,20,11}

בנוסף, הפונקציה תעדכן דרך המשתנה pNewSize את גודל המערך החדש ל-12

#### הערות:

המערכים אינם בהכרח באותו הגודל.

סדר האיברים במערך אינו חשוב.

ניתן להניח של כל מערך מכיל איברים ללא חזרות.

( 2 נק') כתוב את הפונקציה הראשית. הפונקציה יוצרת 2 מערכים דינאמיים ומפעילה את הפונקציה **SymmetricDiff**. לאחר מכן, מדפיסה את המערך המהווה את ההפרש הסימטרי. במידה וההפרש הסימטרי ריק יש להדפיס הודעה מתאימה. בסוף, התוכנית משחררת את כל הזיכרון הדנאמי.

### חלק ג': הקצאה דינמית דו-מימדית (40 נקודות) מענה בקובץ **part3.c**

כתוב פונקציות הבאות :

1. ( 12 נק') **BuildMatrix**(int\* pRow,int\* pColumn) int\*\* הפונקציה תקבל מצביעים למשתנים המהווים את מימדי המטריצה (pRow מצביע למשתנה המייצג את כמות השורות ו pColumn מצביע למשתנה המהווה את כמות העמודות). הפונקציה תקלוט את מספר השורות ומספר העמודות, תקצה זיכרון בהתאם ותמלא את המטריצה בערכים כקלט מהמשתמש ותחזיר את המצביע .

2. ( 5 נק') **PrintMatrix**(int\*\* matrix,int row,int column) void. הפונקציה תקבל מערך דו-מימדי דינאמי, את כמות השורות וכמות העמודות. הפונקציה תדפיס את איברי המטריצה בתצוגת טבלת דו-ממדית .  
 למשל,

1 2 4 7

8 7 6 1

3. ( 13 נק') **TransposeMatrix**(int\*\* matrix,int row,int col,int\* pRow,int\* pCol) int\*\*. הפונקציה תקבל מטריצה matrix, מספר שורות row, מספר עמודות col ומצביעים למשתנים המהווים מימדים של המטריצה המשוחלפת ( pRow מצביע למשתנה המייצג את כמות השורות ו pColumn מצביע למשתנה המהווה את כמות העמודות). הפונקציה תשחלף את המטריצה,

תעדיכן את מימדי המטריצה בהתאם, תיצור את המטריצה המשוחלפת ותחזיר אותה (את המטריצה החדשה, המשוחלפת).

למשל,

עבור דוגמה הקודמת המטריצה המשוחלפת שתודפס:

|   |   |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 4 | 6 |
| 7 | 1 |

(10 נק') כתוב פונקציה ראשית אשר תשתמש בכל הפונקציות הנ"ל כדי לבצע פעולות הבאות :יוצרת מטריצה של מספרים שלמים ומדפיסה אותה. לאחר מכן, התוכנית יוצרת ומדפיסה את המטריצה המשוחלפת.

בסוף, התוכנית תשחרר את כל הזיכרון הדנאמי שהוקצה לה.

**חלק ד': הקצאה דינמית עבור מחרוזת (40 נקודות) מענה בקובץ part4.c**

האורך של כל מחרוזת הוא לכל היותר 20 תווים. במערך הדינמי של מחרוזות יש להקצות זיכרון באופן מדויק.

✓ ( 20 נק' ) כתבו פונקציה

```
void Init(char ***names, int **grades, int *size)
```

הפונקציה מאתחלת באמצעות קלט מהמשתמש מערך של שמות, ומערך של ציונים. שני המערכים באותו גודל, size. בכל פעם שמוסיפים נתונים למערכים יש לוודא שהם חוקיים. אם המשתמש הקליד נתון שאינו חוקי, יש להדפיס הודעת שגיאה, ולבקש נתון חלופי. ציון חוקי הוא שלם בין 0 ל-100. שם חוקי מקיים את התנאים הבאים:

- מתחיל באות לטינית גדולה.
  - כל התווים, פרט לראשון, הם אותיות לטיניות קטנות.
  - לא מופיע כבר במערך. (אסור שהמערך יכיל פעמיים את אותו שם).
- דוגמא לריצת הפונקציה:

```
Enter number of students: 3
Enter name: Ronit
Enter grade: 79
Enter name: yosi
Bad name. Try again: YOSI
Bad name. Try again: Y051
Bad name. Try again: Yosi
Enter grade: 93
Enter name: Ronit
Bad name. Try again: Efrat
Enter grade: 888
```

Bad grade. Try again: 88

✓ ( 10 נק') כתבו פונקציה

**int Find(char \*\*names, int \*grades, int size, char \*name)**

הפונקציה מקבלת כפרמטרים מערך של שמות, ומערך של ציונים, וכן את הגודל של המערכים הללו. בנוסף, היא מקבלת כפרמטר שם של סטודנט. הפונקציה מוצאת את המיקום של הסטודנט במערך השמות, ומחזירה את הציון שלו. אם הסטודנט לא מופיע במערך, יוחזר 1-.

למשל, אם המערכים names, grades הם כמו בדוגמא הקודמת, size=3, name="Efrat", אז הערך המוחזר הוא 88.

✓ ( 5 נק') כתבו פונקציה

**void FreeAll(char \*\*\*names, int \*\*grades, int size)**

הפונקציה משחררת בצורה מסודרת את כל הזיכרון של המערכים.

( 5 נק') כתבו את הפונקציה הראשית שבעזרת פונקציות שיצרתם, בונה מערכים, קולטת שם הסטודנט, מדפיסה את התוצאה ובסוף משחררת את כל הזכרון.

בהצלחה !