

## **תרגיל 4 – מוצבים והקצאה דינמית**

**הגשה עד 4/1/26**

### **הוראות הגשה:**

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראית, גב' **סבטלנה רוסין**,  
במייל: [sceassian2016@gmail.com](mailto:sceassian2016@gmail.com).
2. יש להכין ולהגיש את התרגילים באופן עצמאי. עבודה משותפת אסורה ותיענש בחומרה.
3. להציג רק תכניות שעוברות קומpileציה על מהדר שפת C **CLion**.
4. ההגשה היא של קבצי הקוד (קובץ C) בלבד. יש ליצור 3 קבצים: *c* – *part3.c* ו- *part2.c*, *part1.c*, *lib* – *part3.c*, *part2.c*, *part1.c*.  
לכוץ את כל הקבצים לקובץ אחד בפורמט RAR או ZIP, ולהציג רק קובץ זה.
5. עבודה שתוגש בפורמט אחר מ-C לא תיבדק.
6. הארוכות ינתנו אך ורק במקרים חריגים (מילואים, אבל על קרובים ומהלה חריפה) ובצרכו אישורים מתאימים. כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה האחראי על התרגיל לפני موعد ההגשה.
7. ההגשה היא עד 4/1/26 בשעה 23:50. הגשה מאוחרת אפילו בדקה לא תתקבל. (המערכת חוסמת את אפשרות הגשה). קחו זאת בחשבון ותכננו את זמנכם בהתאם.
8. מותר 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 חיובי או א-שלילי, או בטווח מסוים – אלא אם כן נאמר אחרת בשאלת עצמה.
11. אחרי כל הדפסה יש לבצע ירידת שורה.
12. בתרגיל יש להשתמש בספריות *stdio.h,stdlib.h,string.h* בלבד!
13. יש להקפיד על תכונות נכונות:
  - א. כל הערכים שהם קבועים, ( מבחינה לוגית הם לא אמורים לשנתנות), חייבים להיות מוגדרים כ: *const* ,*define* ,*enum*, בהתאם לצורך.
  - ב. יש לרשום הערות .
  - ג. יש להקפיד על הוצאות!!! כתוב נכון וקריא! ושמות משמעותיים!
  - ד. יש לנסות וליעיל את הקוד והתוכנית ככל שניתן.
  - ה. לפני בקשת קלט (*scanf*) יש להדפיס למשתמש הוראה (*printf*) איזה קלט מבוקש.
  - ו. יש להקפיד על מוסכਮות התכונות הנכון (שמות כמו שצריך וכו').

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

בהצלחה ☺

## **10 נק' חלק א' – part1.c**

### **שאלה 1 (10 נקודות)**

החלק השבור של מספר ממשי הוא הספרות של המספר מימין לנקודה העשרונית. כלומר מספר בין 0 ל-1. **החלק השלים** של מספר ממשי הוא הספרות משמאל לנקודה העשרונית. לדוגמה, עבור 123.4567 החלק השלים הוא 123, והחלק השבור הוא 0.4567. כתוב פונקציה `sums` מקבלת כפרמטרים מערך ממשי וגודלו, הפונקציה מחזירה את הסכום של החלקים השלמים של אברי המערך (מספר שלם), ואת הסכום של החלקים השבריים של אברי המערך (מספר ממשי). הפונקציה לא מדפסה, ולא משנה את אברי המערך. כתוב את הפונקציה **וראשית** בתכנית. הפונקציה קולעת מה משתמש מערך של ממשים וגודלו, מעבירה אותו לפונקציה `sums`, ומדפסה את שני הערכים שחושו. לדוגמה, אם המערך הוא {7.777, 9.9, 8.88, 2.557} אז התכנית תדפיס 24 ו-24.

הערות:

- ניתן להניח ללא בדיקה כי כל המספרים בקלט אינם שליליים.
- מותר להוסיף פרמטרים לפונקציה.

## **40 נק' חלק ב' – part2.c**

### **שאלה 2 (15 נקודות)**

כתב פונקציה `char* CreateString(char* string1, char* string2)` מקבלת 2 מחרוזת ויוצרת מחרוזת חדשה זהה למחרוזת הראשונה עם התוספת הבאה : לאחר כל תו מופיע המחרוזת השנייה. יש להזכיר למחרוזת החדשה כמהן מודיעות של זיכרון. אין לשנות את המחרוזת המקורית.

למשל, הפונקציה מקבלת מחרוזות :

Chocolate 123

הפונקציה, תחזיר מחרוזת החדשה

C123h123o123c123o123l123a123t123e123

### **שאלה 3 (25 נקודות)**

כתב פונקציה **(int \*Start(int \*big, int big\_size, int \*small, int small\_size, int psize)**.  
 הפונקציה מקבלת כפרמטרים שני מערכים של שלמים, big ו-small, וגדליםם. היא יוצרת ומחזירה מערך המכיל את כל האינדקסים במערך big שבהם מתחיל המערך small. באמצעות הפרמטר size היא מחזירה את הגודל של המערך המוחזר. הפונקציה לא מדפיסה דבר, ולא משנה את המערכים big ו-small. לדוגמה, אם  $\{5, 17, 5, 17, 5, 17, 55, 17, 5\}$  = big אז יוחזר המערך  $\{1, 3, 17, 5, 17\}$ . אם big לא מכיל את small אז יוחזר NULL.

כתב את הפונקציה הראשית. הפונקציה קולטת 2 מחרוזות ומעבירה אותן לפונקציה CreateString ומדפיסה את התוצאה, לאחר מכן קולטת מהמשתמש את גודליםם של שני מערכים של שלמים, קולטת את אבריהם, מעבירה אותן כפרמטרים לפונקציה Start, ומדפיסה את המערך המוחזר. בסוף הפונקציה משחררת את כל הזכרון.

### **50 נק', חלק ג' – part3.c**

### **שאלה 4 (25 נקודות)**

כתב פונקציה **(void Minor(int \*\*\*pArr, int\*pSize, int line, int col)**.  
 הפונקציה מקבלת כפרמטר מצביע לערך דו-מימדי של שלמים בגודל size\*size, כאשר size הוא מצביע לsize, ואינדקסים של שורה ועמודה. היא מעדכנת מערך על-ידי מחיקת שורה אחת ועמודה אחת, שהאינדקסים שלhn התקבלו כפרמטרים. הפונקציה לא מדפיסה שום דבר. אם לפחות אחד מבין הפרמטרים col, line או col אינם בטוויה, לא מתבצע עדכון. לדוגמה, עבור  $line=1, col=3$  והמטריצה הבאה

123	987	111	999	12
<b>234</b>	<b>876</b>	<b>222</b>	<b>888</b>	<b>34</b>
345	765	333	777	56
456	654	444	666	78
567	543	555	<b>555</b>	90

תעדכן ל:

123	987	111	12
345	765	333	56
456	654	444	78
567	543	555	90

**שאלה 5 (25 נקודות)**

כתב פונקציה בשם **char\*\* Decompose(char\* str,int\* pCount)** . הfonקציה מקבלת כפרמטר מחרוזת. היא יוצרת ומחזירה מערך מצביעים שכלי איבר בו הוא מילה במחוזת המקורית. באמצעות פרמטר Count היא מחזירה את כמות המילים (מצביעים) . רוח הואתו מפheid בין מילים במחוזת. הfonקציה לא מדפסה, ולא משנה את המחרוזת.

לדוגמא, אם מחרוזת הקלט היא "one two three four five" אז הfonקציה תיצור ותחזיר את המערך מצביעים הבא: {"one", "two", "three", "four", "five"} , בגודל 5.

כתב פונקציה בשם **void SortWords(char\*\* text,int count)** . הfonקציה מקבלת כפרמטר מערך מצביעים וגודל המערך וממיינת אותו לפי האלף-בית. ,

כלומר המערך יתעדכן ל {"five", "four", "one", "three", "two"}.

כתב פונקציה בשם **void PrintWords(char\*\* text,int count)** . הfonקציה מקבלת כפרמטר מערך מצביעים וגודל המערך ומדפסה את המילים.

הfonקציה **ראשית** קולטת גודל המטריצה, יוצרת אותה, קולטת את אבריה, את מספר שורה ומספר עמודה, קוראת לפונקציה Minor ומדפסה את מטריצה המעודכנת.

לאחר מכן הfonקציה קולטת מהמשתמש מחרוזת בגודל 80, מעבירה אותה כפרמטר לפונקציה Decompose, ומדפסה את הערך המוחזר ( מערך מצביעים) , ממינית ומדפסה שוב. בסוף הfonקציה תשחרר את כל הזיכרון .

**בצלחה !!**