

**תרגיל 4 – מצביעים והקצאה דינמית****הגשה עד 4/1/26****הוראות הגשה:**

1. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראית, גב' **סבטלנה רוסיין**, במייל: [sceassign2016@gmail.com](mailto:sceassign2016@gmail.com).
2. יש להכין ולהגיש את התרגילים באופן עצמאי. עבודה משותפת אסורה ותיענש בחומרה.
3. להגיש רק תכניות שעוברות קומפילציה על מהדר שפת **CLion C**.
4. ההגשה היא של קבצי הקוד (קובץ `.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. עבודה שתוגש בפורמט אחר מ-**C** לא תיבדק.
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 האחראי על התרגיל לפני מועד ההגשה.
7. ההגשה היא עד 4/1/26 בשעה 23:50. הגשה מאוחרת אפילו בדקה לא תתקבל. (המערכת חוסמת את אפשרויות ההגשה). 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 יהיה חיובי או א-שלילי, או בטווח מסוים – אלא אם כן נאמר אחרת בשאלה 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 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 😊

### part1.c – 10 נק', חלק א'

**שאלה 1 (10 נקודות)**

**החלק השבור** של מספר ממשי הוא הספרות של המספר מימין לנקודה העשרונית. כלומר מספר בין 0 ל-1. **החלק השלם** של מספר ממשי הוא הספרות משמאל לנקודה העשרונית. לדוגמא, עבור 123.4567 החלק השלם הוא 123, והחלק השבור הוא 0.4567.

כתוב פונקציה sums המקבלת כפרמטרים מערך ממשי וגודלו, הפונקציה מחזירה את הסכום של החלקים השלמים של אברי המערך (מספר שלם), ואת הסכום של החלקים השבורים של אברי המערך (מספר ממשי). הפונקציה לא מדפיסה, ולא משנה את אברי המערך.

כתוב את הפונקציה **הראשית** בתכנית. הפונקציה קולטת מהמשתמש מערך של ממשיים וגודלו, מעבירה אותם לפונקציה sums, ומדפיסה את שני הערכים שחושבו.

לדוגמא, אם המערך הוא {9.9, 8.88, 7.777} אז התכנית תדפיס 24 ו-2.557.

הערות:

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

## part2.c – 'חלק ב'

**שאלה 2 (15 נקודות)**

כתוב פונקציה `char* CreateString(char* string1, char* string2)` המקבלת 2 מחרוזת ויוצרת מחרוזת חדשה זהה למחרוזת הראשונה עם התוספת הבאה : לאחר כל תו מופיע המחרוזת השנייה. יש להקצות למחרוזת החדשה כמות מדויקת של זיכרון. אין לשנות את המחרוזת המקורית.

למשל, הפונקציה תקבל מחרוזות :

Chocolate	123
-----------	-----

## הפונקציה, תחזיר מחרוזת החדשה

C123h123o123c123o123l123a123t123e123

### שאלה 3 (25 נקודות)

כתוב 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`, וגדליהם. היא יוצרת ומחזירה מערך המכיל את כל האינדקסים במערך `big` שבהם מתחיל המערך `small`. באמצעות הפרמטר `psize` היא מחזירה את הגודל של המערך המוחזר. הפונקציה לא מדפיסה דבר, ולא משנה את המערכים `big` ו-`small`. לדוגמא, אם `small = {17,5,17}`, `big = {5,17,5,17,5,17,55,17,5}`, אז יוחזר המערך `{1,3}`. אם `big` לא מכיל את `small` אז יוחזר `NULL`.

כתוב את הפונקציה הראשית. הפונקציה קולטת 2 מחרוזות ומעבירה אותם לפונקציה `CreateString` ומדפיסה את התוצאה, לאחר מכן קולטת מהמשתמש את גודליהם של שני מערכים של מספרים שלמים, קולטת את אבריהם, מעבירה אותם כפרמטרים לפונקציה `Start`, ומדפיסה את המערך המוחזר. בסוף הפונקציה משחררת את כל הזכרון.

### 50 נק', חלק ג' – part3.c

### שאלה 4 (25 נקודות)

כתוב פונקציה `void Minor(int ***pArr, int*pSize, int line, int col)`. הפונקציה מקבלת כפרמטר מצביע למערך דו-מימדי של שלמים בגודל `size*size`, כאשר `pSize` הוא מצביע ל-`size`, ואינדקסים של שורה ועמודה. היא מעדכנת מערך על-ידי מחיקת שורה אחת ועמודה אחת, שהאינדקסים שלהן התקבלו כפרמטרים. הפונקציה לא מדפיסה שום דבר. אם לפחות אחד מבין הפרמטרים `line`, `col` אינם בטווח, לא מתבצע עדכון. לדוגמא, עבור `line=1`, `col=3` והמטריצה הבאה

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

תעודכן ל:

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

### שאלה 5 (25 נקודות)

כתוב פונקציה בשם **char\*\* Decompose(char\* str,int\* pCount)** . הפונקציה מקבלת כפרמטר מחרוזת. היא יוצרת ומחזירה מערך מצביעים שכל איבר בו הוא מילה במחרוזת המקורית. באמצעות פרמטר pCount היא מחזירה את כמות המילים (מצביעים) . רוח הוא תו מפריד בין מילים במחרוזת. הפונקציה לא מדפיסה, ולא משנה את המחרוזת.

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

כתוב פונקציה בשם **void SortWords(char\*\* text,int count)** . הפונקציה מקבלת כפרמטר מערך מצביעים וגודל המערך וממיינת אותו לפי האלף-בית. , כלומר המערך יתעדכן ל {"five","four","one","three","two"}

כתוב פונקציה בשם **void PrintWords(char\*\* text,int count)** . הפונקציה מקבלת כפרמטר מערך מצביעים וגודל המערך ומדפיסה את המילים.

הפונקציה **הראשית** קולטת גודל המטריצה, יוצרת אותה, קולטת את אבריה, את מספר שורה ומספר עמודה, קוראת לפונקציה Minor ומדפיסה את מטריצה המעודכנת.

לאחר מכן הפונקציה קולטת מהמשתמש מחרוזת בגודל 80, מעבירה אותה כפרמטר לפונקציה Decompose, ומדפיסה את הערך המוחזר ( מערך מצביעים) , ממיינת ומדפיסה שוב.

בסוף הפונקציה תשחרר את כל הזיכרון .

**בהצלחה !!**