סמסטר ב', תשפ"ב, אפריל 2022

מבוא למדעי המחשב

# עבודת בית 4

## מצביעים, הקצאת זיכרון דינמית

המחלקה להנדסת תוכנה
המכללה האקדמית להנדסה ע"ש סמי שמעון

# הנחיות

**מועד פרסום:** 11.04.2022 **מועד הגשה:** 01.05.2022

**מתרגלת אחראית:** הדר כוכבי

## מועד הגשה

1. ההגשה היא עד סוף יום ההגשה, כלומר עד השעה 23:59 באותו היום. אל תחכו לרגע האחרון. תכננו את זמנכם בהתאם. הגישו לפני.
2. איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו. בכל מקרה לא יהיה ניתן להגיש מעבר ל-2 ימי איחור ממועד ההגשה דלעיל.
3. במקרים חריגים בלבד ניתן לפנות למרצה לאישור הגשה באיחור מעבר ליומיים הנ"ל.

## אופן הגשה

4. קראו היטב את השאלות. עליכם לענות על כל השאלות בעבודה זו.
5. הגשת העבודה תהיה דרך אתר הקורס במודל בלבד. הגשת העבודה היא ביחידים.
6. כיצד להגיש?
   a. עבור כל משימה יש ליצור קובץ C עם סיומת c. חובה לתת את השמות הבאים לקבצים ex1.c, ex2.c, ex3.c, ex4.c.
   b. חובה לכווץ את כל הקבצים יחדיו לקובץ אחד ויחיד בפורמט RAR או ZIP, ולהגיש רק אותו.
   c. יש להגיש רק תוכניות שעוברות קומפילציה תקינה ללא שגיאות על קומפיילר שפת C Visual Studio 2019.
7. בתחילת כל קובץ יש להוסיף את התיעוד הבא. יש לשנות את השם לשם שלכם ואת תעודת הזהות לתעודת הזהות שלכם.

```
// Assignment: 2

// Author: Israel Israeli, ID: 01234567
```

## שאלות

8. שאלות לגבי העבודה יש לשאול **בפורום באתר המודל של הקורס** או בשעות קבלה של המתרגל/ת האחראי/ת בלבד. אין לשלוח שאלות במייל לא למתרגלת האחראית ולא למתרגלים/מרצים אחרים.
9. ניתן לשאול שאלות הבהרה ומיקוד על המשימות שבעבודה במידה ומשימה מסוימת לא ברורה. לא ניתן לשאול על הפתרונות שלכם. לדוגמא, לא ניתן לשאול האם הפתרון שלי נכון, לא ניתן לשאול למה הפתרון לא עובד, וכדומה.

## קוד

10. ניתן להשתמש בכל החומר שלמדנו ובכל החומר שנלמד עד מצביעים (כולל) אך לא בחומר שנלמד בהרצאות שלאחר מכן.
11. ניתן להשתמש בספריית stdio.h, בספריית assert.h ובספריית stdbool.h בלבד. אסור להשתמש בספריות אחרות. בפרט, **אסור להשתמש בספריה string.h.**
12. הדבר החשוב ביותר הוא שהקוד שלכם יעבוד באופן תקין / נכון. יש לנסות ולייעל את הקוד במקומות בהם ישנה אפשרות להתייעלות.
13. יש להקפיד על תכנות נכון. הערכים שהם קבועים (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: define, const או enum, בהתאם לצורך. יש לכתוב **הערות באנגלית בלבד**, יש להקפיד על הזחות, כיתוב נכון/קריא, שמות משמעותיים וכיוצא באלו.
14. בכל המשימות, אין להניח כי פונקציות הקצאה הזיכרון בהכרח מוצאות זיכרון. יש לטפל גם במקרים בהם לא נמצא זיכרון ע"י malloc, calloc ו realloc. זאת ע"י זיהוי המקרים, הדפסת הודעת שגיאה ויציאה מהתוכנית עם קוד יציאה המעיד על כישלון.

## שונות

15. סך הניקוד עבור משימות 1-3 הוא 57. עבור משימה 4 הניקוד הוא 43.
16. בכל המשימות בעבודה זו הניחו כי הקלט מקיים את כל ההנחות הכתובות במשימה. כלומר, **אינכם נדרשים לבדוק שהקלט מקיים את ההנחות** – בין אם מדובר כארגומנט/פרמטר לפונקציה ובין אם מדובר בקלט מהמשתמש.
17. באפשרותכם לכתוב ולהשתמש בפונקציות עזר, במידת הצורך.

**בהצלחה!**

# עבודת בית 4

## משימה 1: צמצום מחרוזת (21 נקודות)

כתבו פונקציה **char* buildString()** המבצעת את הפעולות הבאות:

1. קולטת מחרוזת מהמשתמש (גודל המחרוזת יהיה לכל היותר 80 תווים).
2. מקצה זיכרון בהתאם לכמות התווים שהמשתמש הזין.
3. קולטת לתוך הזיכרון שהוקצה את הערכים שהמשתמש הזין.
4. מחזירה את כתובת המערך.

כתבו פונקציה **void freeString(char** str)** המקבלת כתובת של המצביע למערך דינאמי str ומשחררת את הזיכרון שהמערך תופס.

כתבו פונקציה **char* reduceString(char* str1, char* str2)** המקבלת שני מצביעים למחרוזות ומבצעת את הפעולות הבאות:

1. מוחקת את כל המופעים של str1 (במידה וישנם) מהמחרוזת str2.
2. מייצרת מערך דינאמי חדש str3 אשר יחזיק את המחרוזת לאחר המחיקה.
3. מחזירה את כתובת המערך.

- שימו לב- הפונקציה לא משנה את str1, str2.

**דוגמאות:**

str1 = “abcd”
str2 = “Xyabczewrrr**abcd**kllabx”
str3 = “Xyabczewrrrkllabx” (= Xyabczewrrr~~abcd~~kllabx)

str1 = “aaa”
str2 = “**aaa**aax**aaaaaa**ay**aaa**aa”
str3 = “aaxayaa” (= ~~aaa~~aax~~aaaaaa~~ay<u>aaa</u>aa)

str1 = “ababab”
str2 = “xyz**ababab**ababz”
str3 = “xyzababz” (= xyz~~ababab~~ababz)

התכנית (main) מבצעת את הפעולות הבאות:

1. התוכנית יוצרת 2 מחרוזות באמצעות **buildString**.
2. מפעילה את **reduceString**.
3. מדפיסה את המחרוזת החדשה.
4. משחררת את הזיכרון.

# משימה 2: עדכון מחרוזת (18 נקודות)

כתבו פונקציה **void printString(char* str)** המקבלת מחרוזת ומדפיסה אותה.

כתבו פונקציה **char* changeString(char *str)** המקבלת מחרוזת המורכבת ממספרים חיוביים, אותיות קטנות ורווחים.

<u>הפונקציה מבצעת את הפעולות הבאות:</u>

1. הפונקציה תיצור מחרוזת חדשה newStr מהמחרוזת str עם השינוי הבא:
   a. כל ספרה תוחלף כך: הספרה 2 תוחלף ב22, הספרה 3 תוחלף ב333.
   b. הספרה 0 תמחק.
   c. רצף רווחים יוחלף בסימן מינוס (-).
   d. רצף של אותיות יוחלף בכוכבית (*).
2. הפונקציה מחזירה את המחרוזת החדשה.

**דוגמא:**

str = “42  qqq231  hh 425 abc  1023”
new_str = “444422-*223331-*-44442255555-*-122333”

- ניתן להשתמש בפונקציה **buildString** ממשימה 1 ליצירת המחרוזת str.

<u>התכנית (main) מבצעת את הפעולות הבאות:</u>

1. התוכנית יוצרת מחרוזת באמצעות **buildString** ממשימה 1.
2. מפעילה את **changeString**.
3. מדפיסה את המחרוזת החדשה.
4. משחררת את הזיכרון.

# משימה 3: מטריצת מינור (18 נקודות)

כתבו פונקציה **int** minor(int** matrix, int size, int i, int j)** המקבלת מטריצה ריבועית של מספרים שלמים, גודל המטריצה, מספר שורה ומספר עמודה.

<u>הפונקציה מבצעת את הפעולות הבאות:</u>

1. הפונקציה מקצה זיכרון למטריצת המינור.
2. מחשבת <u>מטריצת מינור של מטריצה ריבועית נתונה</u>.
3. מחזירה מצביע למטריצה החדשה.

כתבו פונקציה **void freeMatrix(int*** matrix, int row)** המקבלת מצביע למערך דו-מימדי וכמות השורות, ותשחרר את כל הזיכרון שהמטריצה תפסה.

כתבו פונקציה **void printMatrix(int** matrix, int rows, int columns)** המקבלת מערך דו-מימדי דינאמי, כמות השורות והעמודות. הפונקציה תדפיס את איברי המערך בתצוגת טבלה דו-מימדית.

**דוגמא:**

עבור מטריצה 2x2:

1 2

8 7

<u>התכנית (main) מבצעת את הפעולות הבאות:</u>

1. קולטת מהמשתמש מטריצה ריבועית A
   - מדפיסה את המטריצה הריבועית שנקלטה.
2. קולטת אינדקס שורה i ואינדקס עמודה j.
   - מדפיסה את אינדקס השורה ואינדקס העמודה שנקלטו.
3. מפעילה את הפונקציה minor לחישוב מטריצת המינור $A_{ij}$.
   - מדפיסה את מטריצת המינור שהתקבלה.
4. משחררת את הזיכרון שהוקצה.

בשאלה זו יש להשתמש במצביעים והקצאות דינאמיות – אין להשתמש במערכים (בפרט, אין להשתמש ב [ ]).

**דוגמא:**

עבור מטריצה A:

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 3 & 0 & 5 \\ -1 & 9 & 11 \end{pmatrix}$$

מטריצת המינור של המטריצה A לפי שורה 1 ועמודה 0 היא:

$$\begin{pmatrix} 3 & 0 \\ -1 & 9 \end{pmatrix}$$

**דוגמא:**

עבור מטריצה A

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 3 & 0 & 5 \\ -1 & 9 & 11 \end{pmatrix}$$

מטריצת המינור שלה לפי שורה 2 ועמודה 1 היא:

$$\begin{pmatrix} 1 & 4 \\ -1 & 9 \end{pmatrix}$$

**דוגמת הרצה של התכנית:**

```
Enter size of matrix: 3
Enter A[0][0]: 1
Enter A[0][1]: 2
Enter A[0][2]: 3
Enter A[1][0]: 4
Enter A[1][1]: 5
Enter A[1][2]: 6
Enter A[2][0]: 7
Enter A[2][1]: 8
Enter A[2][2]: 9
1 2 3
4 5 6
7 8 9
Enter i: 0
Enter j: 2
You entered: i=1, j=0

The minor matrix with respect to 1 and 0 is:
4 5
7 8
```

# משימה 4 (43 נקודות)

משימה זו מחולקת לתתי משימות.
מותר להשתמש בפונקציות שכתבתם במשימות קודמות. לדוגמא, ניתן להשתמש בפונקציה שכתבת במשימה 4.1 כאשר אתם פותרים את משימה 4.7.

## משימה 4.1:

כתבו פונקציה **void BuildMatrix(int*** matrix, int row, int column)** המקבלת מצביע למערך דו-מימדי דינאמי, כמות שורות ועמודות.

<u>הפונקציה תבצע את הפעולות הבאות:</u>

1. מקצה זיכרון בהתאם לעמודות והשורות.
2. קולטת מהמשתמש ערכים וממלאה את המטריצה לפיהם.

## משימה 4.2:

כתבו פונקציה **void addColumn(int*** matrix, int rows, int* columns, int* newCol, int row)**

<u>הפונקציה מקבלת:</u>

1. מצביע למערך דו-מימדי דינאמי (מטריצה), ואת כמות השורות ועמודות של המערך.
2. מערך חד-מימדי **newCol** בגודל row.

<u>הפונקציה מבצעת את הפעולות הבאות:</u>

1. הפונקציה מוסיפה את העמודה **newCol** למטריצה בתור העמודה האחרונה שלה.
2. אם קיימת אי התאמה במימד הרלוונטי של המטריצה (למשל- row יותר גדול מrows), יש להציג הודעה מתאימה.
3. הפונקציה מעדכנת את הפרמטר column לפי מספר העמודות החדש.

**דוגמא:**

מטריצה מקורית:

1 2 4 7
8 7 6 1

new_col = 1 9

מטריצה לאחר העדכון:

1 2 4 7 **1**
8 7 6 1 **9**

<u>התוכנית (main) מבצעת את הפעולות הבאות:</u>

1. קולטת מהמשתמש ערכים לבניית מטריצה ויוצרת אותה.
2. קולטת ממנו את **row** ו- **newCol.**

3. מפעילה את **addColumn**.
4. מדפיסה את המטריצה החדשה שנוצרה.

## משימה 4.3:

כתבו פונקציה **void removeColumn(int*** matrix, int rows, int* columns, int colNum)**

<u>הפונקציה מקבלת:</u>

3. מצביע למערך דו-מימדי דינאמי (מטריצה), ואת כמות השורות ועמודות של המערך.
4. מספר העמודה אותה יש למחוק.

<u>הפונקציה מבצעת את הפעולות הבאות:</u>

4. הפונקציה מוחקת את העמודה מספר colNum.
5. אם קיימת אי התאמה במימד הרלוונטי של המטריצה (למשל- colNum חורג מגבולות המטריצה), יש להציג הודעה מתאימה.
6. הפונקציה מעדכנת את הפרמטר column לפי מספר העמודות החדש.

**דוגמא:**

מטריצה מקורית:

1 2 4 7
8 7 6 1

colNum = 2

מטריצה לאחר העדכון:

1 2 7
8 7 1

<u>התוכנית (main) מבצעת את הפעולות הבאות:</u>

1. קולטת מהמשתמש ערכים לבניית מטריצה ויוצרת אותה.
2. קולטת את **colNum.**
3. מפעילה את **addColumn**.
4. מדפיסה את המטריצה החדשה שנוצרה.

## משימה 4.4:

כתבו פונקציה **int** multiplyMatrixes(int** mat1, int** mat2, int m, int n, int k)**

הפונקציה מקבלת מטריצה mat1 עם m שורות ו- n עמודות, ומטריצה mat2 עם n שורות ו- k עמודות.

<u>הפונקציה מבצעת את הפעולות הבאות:</u>

1. הפונקציה יוצרת מטריצה mat3 בגודל m שורות וk עמודות.

2. המטריצה mat3 היא הכפל של mat1 ו-mat2.
3. מחזירה את mat3.

כפל מטריצות:

$$\begin{matrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{matrix} \times \begin{matrix} b_1 & b_2 \\ b_4 & b_5 \\ b_7 & b_8 \end{matrix} =$$

$$\begin{matrix} (a_1 * b_1 + a_2 * b_4 + a_3 * b_7) & (a_1 * b_2 + a_2 * b_5 + a_3 * b_8) \\ (a_4 * b_1 + a_5 * b_4 + a_6 * b_7) & (a_4 * b_2 + a_5 * b_5 + a_6 * b_8) \end{matrix}$$

**דוגמא:**

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix} \times \begin{matrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{matrix} = \begin{matrix} (1+4+9) & (4+10+18) \\ (4+10+18) & (16+25+36) \end{matrix} = \begin{matrix} 14 & 32 \\ 32 & 77 \end{matrix}$$

<u>התכנית (main) מבצעת את הפעולות הבאות:</u>

1. קולטת מהמשתמש ערכים לבניית 2 המטריצות ויוצרת אותן.
2. מפעילה את **multiplyMatrixes**.
3. מדפיסה את המטריצה החדשה שנוצרה.

## משימה 4.5:

כתבו פונקציה **void transpose(int*** matrix, int* row, int* column)**

הפונקציה מקבלת מטריצה, מספר השורות והעמודות שלה.

<u>הפונקציה מבצעת את הפעולות הבאות:</u>

1. הפונקציה יוצרת מטריצה mat בגודל column עמודות ו-row שורות.
2. משחלפת את המטריצה כך שכל עמודה תהפוך לשורה ולהיפך.
3. משחררת את הזיכרון של המטריצה הקודמת.

**דוגמא:**

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix} \rightarrow \begin{matrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{matrix}$$

<u>התכנית (main) מבצעת את הפעולות הבאות:</u>

1. קולטת מהמשתמש ערכים לבניית המטריצה ויוצרת אותה.
2. מפעילה את **transpose**.
3. מדפיסה את המטריצה החדשה שנוצרה.

# משימה 4.6:

בכניסה למערכת התוכנית תציג למשתמש את התפריט אשר ממנו יבחר את הפונקציה שירצה להפעיל:
את התפריט יש לכתוב בפונקציית main:

1 – Add Column

2 – Remove Column

3 – Multiply Matrixes

4 – Transpose

5 – Exit

ההרצה תעבוד עד לבחירת אופציה 5, אשר תסיים את התוכנית ותודפס ההודעה הבאה:

Have an awesome day! Trilili Tralala

עליכם לדאוג לשחרר את כל הזיכרון שהוקצה במהלך פעולת התוכנית טרם סיומה.

**בהצלחה!**