



מכון טכנולוגי חולון  
Holon Institute of Technology

## מחלקה למדעי המחשב COMPUTER SCIENCE DEPARTMENT

סדנה מתקדמת בתכנות 61108  
סמסטר ב' תשפ"ג

### מטלה 2

מבנים, מערכים דו-ממדיים, רשימות מקושרות, מחרוזות, קבצים

### דרישות חובה והסברים כלליים לתרגיל הגשה:

הדרישות הכלליות המופיעות בסעיף זה, הינם דרישות חובה. אי קיום הדרישות יוביל לפסילת שאלות וחבל.

- תרגיל הגשה זה מורכב מ-4 שאלות.
- לכל שאלה ישנו קובץ 'c' מצורף, המכיל את השלד של התרגיל. אין לשנות את שמות הקבצים.
- למשל: עבור השאלה הראשונה שם הקובץ הוא ex2\_q1.c, שאלה שנייה: ex2\_q2.c וכן הלאה...
- בכל קובץ קיימים מספר פונקציות, חלקם עליכם לממש. אין לשנות את שמות הפונקציות, ואין לשנות את חתימת הפונקציות
- את המימושים יש לכתוב החל מהשורה שלאחר ההערה: // your code
- אין למחוק את הערות אלו
- לפני חלק מהפונקציות מופיעה הערה: // DO NOT CHANGE the following function
- אין לשנות פונקציה המופיעה לאחר הערה כזאת
- לפני חלק מהפונקציות מופיעה הערה: // DO NOT CHANGE from this point
- אין לשנות או להוסיף קוד החל מהערה זו עד סוף הפונקציה
- אין לשנות את שמות הפונקציות שכבר מופיעות בקבצים
- למעט שאלה מס' 1 בה אתם נדרשים להוסיף פונקציה אחת, ושאלה מס' 4.
- אין להוסיף פונקציות חדשות לאף אחד מן התרגילים.
- ניתן להוסיף include במידת הצורך.
- יש להתייחס באופן מלא להערות נוספות אשר מופיעות בקבצי ההרצה, אין למחוק הערות אלה.

## שאלה 1:

נגדיר את שכניו של איבר  $a[i][j]$  במטריצה להיות כל האיברים הסובבים אותו בקו ישר ובאלכסון. לדוגמא:

$$\begin{bmatrix} 7 & 6 & 1 \\ 8 & 3 & 2 \\ 9 & 4 & 0 \end{bmatrix}$$

עבור המטריצה הבאה, לאיבר המרכזי שערכו 3 יש 8 שכנים, עבור האיברים שנמצאים בפינות יש 3 שכנים.

נגדיר מבנה "מספר מעורב" כלומר מספר שלם שאחריו שבר (שערכו קטן מ-1). המבנה מכיל שלושה שדות מטיפוס int: חלק שלם (num), מונה (numerator) ומכנה (denominator).

```
typedef struct fraction
{
    int num, numerator, denominator;
} fraction;
```

כתבו פונקציה בשם matrixAverageNeighbor שמקבלת מטריצה A המיושמת כמערך דו-ממדי סטטי וגדליה. על הפונקציה להקצות מערך דו-ממדי חדש - מטריצה B של איברים מסוג **Fraction** בעלת אותם גדלים כמו המטריצה A כך שכל איבר  $b[i][j]$  שלה יהיה שווה לממוצע של כל שכני האיבר  $a[i][j]$ .

## דוגמה:

$$A: \begin{bmatrix} 5 & 12 & 6 & 8 \\ 4 & 7 & 0 & 9 \\ 13 & 20 & 8 & 2 \\ 18 & 0 & 2 & 6 \end{bmatrix} \rightarrow B: \begin{bmatrix} 7\frac{2}{3} & 4\frac{2}{5} & 7\frac{1}{5} & 5 \\ 11\frac{2}{5} & 8\frac{1}{2} & 9 & 4\frac{4}{5} \\ 9\frac{4}{5} & 6\frac{1}{2} & 5\frac{3}{4} & 5 \\ 11 & 12\frac{1}{5} & 7\frac{1}{5} & 4 \end{bmatrix}$$

שימו לב: עבור המטריצה A ערכי המטריצה B צריכים להיות בדיוק כפי שהם מופיעים בדוגמא. אם איבר של המטריצה B הוא בפועל מספר שלם, אז השדה numerator של המשתנה יהיה 0 וערך השדה denominator שלו יכול להיות שרירותי. יש לצמצם שברים למקסימום ( $\frac{2}{4} \rightarrow \frac{1}{2}$ ).

הפונקציה תחזיר את הכתובת של המטריצה הדינאמית B החדשה.

ניתן להניח שיש בזיכרון מספיק מקום להקצאה.

יש להשתמש בפונקציית עזר בשם neighborFractionAverage אשר מקבלת מערך דו-ממדי סטטי עם קואורדינטות i ו-j של איבר ומחזירה טיפוס מסוג fraction.

יש למלא בהתאמה את שאר הפונקציות המופיעות בקובץ השלד: הקצאת מטריצה דינאמית, הדפסת מטריצה דינאמית ושחרור מטריצה דינאמית.

יש לכתוב את הפונקציה להדפסת המטריצה B לפי ייצוג מסוג double, עם 2 ספרות אחרי הנקודה.

**לשיקולכם:** בשאלה זו מותר להוסיף פונקציית עזר **אחת בלבד** נוספת לשימושכם. מיקום הפונקציה יוגדר בהתאם באזור מוגדר בקובץ השלד.

## שאלה 2:

כתבו פונקציה בשם `createArrayAndList` המקבלת מטריצה כמערך דו-ממדי סטטי וגדליה. הפונקציה מוצאת את כל אחד מאיברי המטריצה שערכם מהווים סדרה חשבונית ביחד עם ערכי ה- $i$  ו- $j$  שלהם (המתחילה ב- $i$  ומסתיימת ב- $a[i][j]$ ).

### **לדוגמא:**

$$\begin{aligned} a[1][2] &= 3 \text{ (הסדרה משמאל לימין: 1, 2, 3)} \\ a[2][4] &= 6 \text{ (הסדרה משמאל לימין: 2, 4, 6)} \\ a[4][2] &= 0 \text{ (הסדרה משמאל לימין: 4, 2, 0)} \end{aligned}$$

על הפונקציה לבנות מערך דינאמי חד-ממדי של רביעיות, וכן רשימה מקושרת **חד כיוונית** של רביעיות. כל רביעייה במערך או ברשימה תכיל את ערכו של איבר המטריצה שעונה לתנאי הנ"ל, את הקואורדינטות  $i$  ו- $j$  (מס' השורה ומס' העמודה) של אותו האיבר במטריצה ואת הערך  $d$  של הסדרה החשבונית. הפונקציה תחזיר את גודל המערך (ששווה לאורכה של הרשימה), ותעביר (by reference) את המערך ואת הרשימה.

### לדוגמא, עבור המטריצה המקורית:

i / j	0	1	2	3	4
0	0	6	5	6	6
1	8	9	5	6	7
2	7	6	5	4	7
3	9	8	1	6	7

### סדרת הרביעיות תהיה:

איבר	0	6	7	4	1
i	0	0	1	2	3
j	0	3	4	3	2
d	0	3	3	1	-1

### **והפונקציה תחזיר 5.**

```
typedef struct four
{
    int i, j, d, value;
} four;
```

```
typedef struct list
{
    four data;
    struct list *next;
} list;
```

למימוש הרביעיות יש להשתמש בטיפוס `struct four` בעל ארבעה שדות נומריים. איברי המערך יהיו מבנים מסוג זה. כל איבר של הרשימה יהיה מורכב משני שדות: נתון ומצביע לאיבר הבא. הנתון הוא `struct four` (כמו איברי המערך).

**יש למלא ולהשתמש בהתאם** בשאר הפונקציות המופיעות בקובץ השלד:

(1) פונקציה בשם `createFour` המקבלת ארבעה מספרים שלמים ומחזירה רביעייה אחת המורכבת מארבעת הפרמטרים שקיבלה, (2) פונקציה להקצאת איבר כלשהו ברשימה מקושרת, (3) פונקציה להדפסת מערך, (4) פונקציה להדפסת רשימה ו- (5) פונקציה לשחרור אובייקטים שהוקצו באופן דינאמי.

יש להתייחס לחתימות הפונקציות כפי שהן כתובות בשלד, ולהנחיות שמופיעות בהן. ניתן להניח שיש בזיכרון מספיק מקום להקצאה, יש להתייחס לכל מקרי הקצה האפשריים (רשימה ריקה וכו'). **אין להוסיף פונקציות נוספות מעבר לפונקציות הקיימות.**

### שאלה 3:

כתבו פונקציית פענוח בשם decode המקבלת מחרוזת אשר מכילה טקסט מוצפן המורכב ממספר מילים, כאשר כל מילה מופרדת ע"י רווח שאינו מוצפן. על הפונקציה לפענח את הטקסט לפי השיטה הבאה:

- יש להמיר כל אות גדולה לאות גדולה אחרת, וכל אות קטנה לאות קטנה אחרת.
- יש להמיר סימנים אחרים (מספרים וכל השאר) לסימנים אחרים.
- רווחים נשארים כפי שהם.

כל מילה מוצפנת בנפרד ללא תלות במילה הקודמת, יש להתייחס למיקום האותיות (לא כולל סימנים), ולהוסיף לערך Ascii של כל אות במילה  $2^i$  בשקילות מודולרית כאשר  $i$  מתחיל מ-0, למעשה מייצג את האינדקס של האות במילה.

סימנים מוצפנים ביחס לכל הטקסט באופן הבא: יש להחסיר 1 מערך ה-Ascii של תו הסימן הראשון, יש להחסיר 2 מערך ה-Ascii של תו הסימן השני וכו'.

הפונקציה תחליף את התוכן המקורי של המחרוזת בתוכן החדש (שמהווה את הטקסט המפוענח) ללא שימוש במערך עזר.

Example:

H	r	(	o		Z	j	e	n	o	#
$+2^0$	$+2^1$	-1	$+2^2$		$+2^0$	$+2^1$	$+2^2$	$+2^3$	$+2^4$	-2
l	t	'	s		A	l	i	v	e	!

יש לבחון את הפונקציה עבור הטקסט המוצפן הבא: (מופיע בשלד התרגיל)

"H gmlw xmq(jo ojascyr vgpz xmqjczyxia0 Vc bmqdn zjh gywv acaf jghdoh 0 np vmnko0 dvlwv3 Mms hd xmq cmj-l lgjv3 H/k fmefq sm acz7"

**ניתן להניח כי אורך כל מילה לא יהיה גדול יותר גדול מ-32. אין להוסיף פונקציות נוספות מעבר לפונקציות הקיימות.**

### שאלה 4:

כתבו פונקציה בשם memoryReport המקבלת מחרוזת filename שתוכן שלה הוא נתיב של קובץ המכיל מספר שורות של הכרזות פשוטות של משתנים בשפת C. על הפונקציה לקרוא את הקובץ ולחשב את כמות הזיכרון שיש להקצות עבור כל משתנה ולהדפיס למסך פירוט בהתאמה. כמו כן, הפונקציה תחזיר את גודל הזיכרון הנדרש הכולל.

file1.txt

```
int x, y;  
char c, *cptr, carray[80];
```

Function print:

x requires 4 bytes  
y requires 4 bytes  
c requires 1 byte  
cptr requires 4 bytes  
carray requires 80 bytes

Function return: 93

כל שורת הצהרה בקובץ מהווה **שורה תקינה** בשפת C וצריכה להכיל:

- אחד משמונת טיפוסים הבאים:  
char, short, int, unsigned int, long, long long, float, double.
- הכרזה אחת או יותר של משתנים לא מאותחלים, מופרדים בפסיקים מהסוגים הבאים:  
משתנים פשוטים או מצביעים או מערכים חד-ממדיים או מערכים של מצביעים חד-ממדיים או מערכים דו-ממדיים.
- לא יהיו מערכי מצביעים דו-ממדיים, מצביעים למצביעים או מערכים תלת-ממדיים וכו'.
- נקודה פסיק לציון סוף שורה.
- רווחים ו-Tabs ללא הגבלה במקומות המתאימים (כמו בשורות הכרזות אמיתיות).

הנחות נוספות:

- גודל מצביע הוא 4 בתים (לפי מערכת הפעלה 32).
- אחרי סימן נקודה פסיק לא יופיעו הצהרות נוספות באותה שורה.

**ניתן להוסיף פונקציות נוספות מעבר לפונקציות הקיימות, יש להצהיר עליהן בהתאם בקובץ השלד**

לנוחיותכם, מצורפים שלושה קבצי הכרזות לבדיקה בצמוד לקובץ השלד.  
יש לשתול אותם בנתיב: c:\temp.  
אין צורך להגיש אותם.

### הוראות נוספות:

1. במידה ושם הפונקציה במסמך אינו תואם את השם שמופיע בקובץ השלד, שם הפונקציה בקובץ השלד היא זאת שקובעת.
2. על התוכניות להיות יעילות ככל האפשר.
3. יש להשתמש בשמות משמעותיים וגם בהערות.
4. יש להקפיד לכתוב בצורה מבנית.
5. אין צורך לבצע בדיקת תקינות קלט וניתן להניח שיש בזיכרון מספיק מקום להקצאה.
6. יש להתייחס לכל מקרי הקצה האפשריים (רשימה ריקה וכו').
7. אין צורך לבצע קליטת נתונים בתרגיל זה.
8. תכנית שלא עוברת קומפילציה לא תתקבל!

**בהצלחה!!!**