המכללה האקדמית להנדסה סמי שמעון

דף תרגילים, מעבדה מס'4
(מחלקה וניהול זיכרון דינמי)

הגדר מחלקה **CMatrixFun**, המתארת מטריצה דינאמית של מספרים ממשיים מסוג **double**, על-ידי שימוש במצביע כפול לסוג הנתון ושני מספרים שלמים המהווים את ממדי המטריצה (מס' השורות ומס' העמודות).

יש לבנות עבור המחלקה את הפונקציות הבאות:

א) בנאי ברירת המחדל ()**CMatrixFun**, שתפקידו ליצור מטריצה מסדר 1x1 ולאפס אותה.

ב) בנאי המקבל שני מספרים שלמים, (int rows, int colums)**CMatrixFun** המהווים את ממדי המטריצה (מס' השורות ומס' העמודות). תפקיד הבנאי הוא ליצור מטריצה לפי הממדים שהתקבלו ולאתחל את ערכי שדות המטריצה בהתאם לממדים אלו, כאשר הערכים יתקבלו ממחולל **אקראי***. בתחום בין **0** ל-**30**.

ג) בנאי מעתיק.

ד) פונקציה הורסת.

ה) פונקציה ()void **Print**, המדפיסה את איברי המטריצה בתצוגת טבלת דו-ממדית.

ו) פונקציה (int row, int column, double newValue)void **SetValue**, המקבלת את מיקום האיבר ואת ערכו החדש. במידה וקיים מיקום זה במטריצה, הערך יתעדכן בהתאם ל-newValue, אחרת יש להציג הודעה מתאימה ואין לנקוט באף פעולה נוספת.

ז) פונקציה (int row, int column)double **GetValue**, המקבלת את מיקום האיבר ומחזירה את הערך המוגדר במיקום זה. במידה ולא קיים מיקום זה במטריצה, תוצג הודעה מתאימה והתכנית תסתיים.

ח) פונקציה ()void **Transpose**: משחלפת את המטריצה כך שכל עמודה תהפוך לשורה וההיפך, לדוגמא :

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

ט) פונקציה (double * ar, int size_ar)void **AddRow**, המקבלת מערך חד-ממדי של מספרים ממשיים ומוסיפה אותו כשורה האחרונה החדשה. על הפונקציה לבדוק את מידת ההתאמה של השורה החדשה לממדי המטריצה הקיימת, במידה וקיימת אי-התאמה בממד הרלוונטי של המטריצה יש להציג הודעה מתאימה ולסיים את הפונקציה.

י) פונקציה (int colNumber)void **RemoveColumn**, המקבלת את מספר העמודה ומורידה אותה מהמטריצה הקיימת. במידה ומספר העמודה אינו תואם את ממדי המטריצה, תוצג הודעה מתאימה והפונקציה תסתיים.

יא) פונקציה (int rowNumber)double * **GetRow**, המקבלת את מספר השורה ומחזירה את הערכים בשורה זו בצורת מערך .במידה ומספר השורה אינו תואם את ממדי המטריצה, תוצג הודעה מתאימה והפונקציה תסתיים.

יב) פונקציה (const CMatrixFun &)CMatrixFun **Multiply**, המקבלת אובייקט מסוג מטריצה (CMatrixFun) והמחזירה אובייקט חדש מסוג **CMatrixFun** אשר התקבל על ידי מכפלה של איברי שתי המטריצות(להלן הסבר קצר-**קישור**)

המכללה האקדמית להנדסה סמי שמעון

יג) פונקציה בוליאנית **CheckArithmeticSequence**() bool, הבודקת האם סדרת הערכים הנמצאים על היקף המטריצה החל מהאיבר 0,0 ועד לאיבר 1,0 בהתקדמות לפי כיוון השעון מהווה טור חשבוני [**הפרש קבוע** בין כל זוג מספרים עוקבים].
למשל מטריצה הבא מקיימת את התנאי הזה:

2	4	6	8
24	8	3	10
22	1	8	12
20	18	16	14

יד) פונקציה **MaxRelocate**() CMatrixFun, המחזירה אובייקט חדש מסוג **CMatrixFun** אשר התקבל בהחלפת שתי שורות ושתי עמודות בלבד על מנת שהאיבר המקסימלי יעבור למיקום 0,0.

לדוגמא:

15	27	8
3	18	25
21	**29**	20
22	25	25
1	17	0

29	21	20
18	3	25
27	15	8
25	22	25
17	1	0

כתוב תכנית אשר יוצרת אובייקטים מהסוג הנ"ל ומפעילה את הפונקציות/מתודות של המחלקה (לצורך כך ניתן לבנות תפריט מתאים).

הערות:

א) על משתני המחלקה להיות פרטיים.

ב) לא ניתן להוסיף משתנים נוספים למחלקה.

ג) ניתן להוסיף פונקציות נוספות לפי הצורך.

```
*/* rand example: guess the number */
#include <iostream>
#include <time.h>
using namespace std;
int main () {
	int iSecret, iGuess;

	srand (time(NULL)); /* initialize random seed: */
	iSecret = rand() % 10 + 1; /* generate secret number between 1 and 10: */
	do {
		cout<<endl<<" Guess the number (1 to 10): ";
		cin>>iGuess;
		if (iSecret<iGuess)
			cout<<" The secret number is lower ";
		else if (iSecret>iGuess)
			cout<<" The secret number is higher ";
	} while (iSecret!=iGuess);
	cout<<"Congratulations!"<<endl;
	return 0;
}
```

בהצלחה!