

סמסטר א' תשע"ג מועד ב
תאריך: 1.3.2013
שעה: 9:00
משך הבחינה: 4 שעות
כל חומר עזר אסור בשימוש



בחינה בקורס: מבוא למדעי המחשב

- יש לענות על כל 5 השאלות.
- בכל השאלות בבחן יש לכתוב פונקציות ייעילות ככל האפשר גם בזמן וגם בזכרון.
- שימרו על סגנון התכנות. הגדרו קבועים היכן שצרכיך וכתבו פונקציות עוזר.
- יש לשחרר את כל שטחי הזיכרון שהוקכו דינאמית כאשר אין בהם צורך יותר.
- אין צורך לבדוק הצלחה של הקצאות דינמיות.
- בכל השאלות בבחן ניתן להניח שהקלט תקין.
- אלא אם נאמר אחרת, ניתן להשתמש (מבלי למשם) בכל הפונקציות הנתונות בעמוד 2 של מבחן זה.
- אין צורך לטעד את הפונקציות.
- יש לכתוב תי'ז ומספר מחברת בראש כל עמוד. עמוד ללא תי'ז בראשו לא יבדק.
- יש לענות על כל שאלה במקום שמיועד לה בגוף השאלה. המחברות הן טויטה בלבד ולא תידקנה.
- שים לב כי הבחינה כילתה 15 עמודים, כולל עמוד זה.

בהצלחה!

רשימת פונקציות ספרייה מותרót לשימוש:

פונקציה	ספרייה
int strlen(char str[])	String.h
void strcpy(char dest[], char src[])	String.h
void strcat(char dest[], char src[])	String.h
int strcmp(char s1[], char s2[])	String.h
int __max(int a, int b)	stdlib.h
int __min(int a, int b)	stdlib.h
double pow(double base, double exp)	math.h
double sqrt(double x)	math.h
double fabs(double x)	math.h
Cout	iostream
Cin	iostream
char cin.get(char& outCh)	iostream
void cin.getline(char outStr[], int max, char delimiter)	iostream

שאלה 1 (20 נקודות):

הנימו שМОודר הקבוע

```
const int MAX_SIZE=100 ;
```

כתבו את הפונקציה

```
int findMulInd(int matNum[ ][MAX_SIZE], int size,
               int reqMul) ;
```

הfonקציה מקבלת כקלט מטריצה ריבועית (בגודל לוגי $\text{size} * \text{size}$) של מספרים שלמים, את size שווה למספר השורות ושווה למספר העמודות במטריצה ומכללה מבוקשת reqMul .

הfonקציה תחשב ותחזיר את סכום האיברים שמכפלת האינדקסים שלהם שווה ל reqMul .

$$mat = \begin{bmatrix} 2 & 3 & 1 & 2 & 5 \\ 4 & 2 & 3 & 8 & 2 \\ 1 & 0 & 9 & 1 & 5 \\ 6 & 2 & 3 & 8 & 2 \\ 10 & 7 & 2 & 3 & 1 \end{bmatrix}$$

לדוגמא, נניח כי נתונה המטריצה

הקריאה לפונקציה `findMulInd(mat, 5, 4)` תחזיר את הערך 18

$$\text{כי : } mat[1][4] + mat[2][2] + mat[4][1] = 2 + 9 + 7 = 18$$

$$\text{ואו : } 4 = 2 * 2 - 1 = 4 = 4 * 1$$

הקריאה לפונקציה `findMulInd(mat, 5, 0)` תחזיר את הערך 34

כי :

$$\begin{aligned} mat[0][0] + mat[0][1] + mat[0][2] + mat[0][3] + mat[0][4] + \\ + mat[1][0] + mat[2][0] + mat[3][0] + mat[4][0] = \\ = 2 + 3 + 1 + 2 + 5 + 4 + 1 + 6 + 10 = 34 \end{aligned}$$

```
int findMulInd(int matNum[ ][MAX_SIZE], int size,  
               int reqMul)
```

שאלה 2 (20 נקודות):

כתבו את הפונקציה :

```
bool equalPartP ( int numArr[ ], int size ) ;
```

הfonקציה מקבלת כקלט מערך numArr של מספרים שלמים ואת גודלו size .
 $\text{size} \geq 2$

הfonקציה מחזירה ערך true אם קיימים אינדקס p במערך ($0 \leq p \leq \text{size} - 2$) קיימים רצופים ססכום שווה, כלומר מתקיים ש :

$$\begin{aligned} \text{numArr}[0] + \text{numArr}[1] + \dots + \text{numArr}[p] &= \\ &= \text{numArr}[p + 1] + \dots + \text{numArr}[\text{size} - 1] = \\ &= \frac{\text{numArr}[0] + \text{numArr}[1] + \dots + \text{numArr}[\text{size} - 1]}{2} \end{aligned}$$

למשל

true equalPartP ([1,2,6,5,2,12],6) ○
 כי $p=3$ קלומר $1+2+6+5=12+2$

true equalPartP ([27,2,6,5,2,12],6) ○
 כי $p=0$ קלומר $27=2+6+5+12+2$

false equalPartP ([3,2,6,5,2,12],6) ○
 כי לא קיים p מותאים.

עמוד 6 מתוך 15

מספר מהברת:

ת'ז:

```
bool equalPartP (int numArr[], int size)
```

שאלה 3 (20 נקודות):

נתון הקבוע הבא :

```

const int MAX_NAME_LEN = 100;

ונתונה הגדירה הבאה לייצוג עובד בcplusplus :

struct employee{
    char familyName[MAX_NAME_LEN];      // שדה המכיל את שם המשפחה
    char firstName[MAX_NAME_LEN];        // שדה המכיל את השם הפרט
    int empNum;                         // שדה המכיל את מספר העובד
    int salary;                          // שדה המכיל את משכורת העובד

};

typedef struct employee EMPLOYEE;

```

כתבו את הפונקציה הבאה :

```

int* findSalary(EMPLOYEE empleos[], int size,
                 int reqSalary, int &new_size);

הפונקציה מקבלת מערך empleos של עובדים ואת גודלו size ובנוסף משכורת סך
מボוקשת .reqSalary.

הפונקציה תקצה ותחזיר מערך ובו מספרי העובדים (השדה empNum) אשר
המשכורת שלהם שווה לפחות לסכום .reqSalary. גודל המערך החדש יוחזר
במשתנה הפלט .new_size

```

```
int * findSalary(EMPLOYEE emplos[], int size,  
                  char ReqFamily, int &new_size)
```

שאלה 4 (20 נקודות):

כתבו את הפונקציה

```
void findLetters(char str1[], char str2[]);
```

הfonקציה מקבלת קלט שני מחרוזות str1 ו-str2 (העשויות להיות ארכות מאוד) המכילות אותיות lower case (קטנות) ורווחים ומדפיסה את כל האותיות שופיעות לפחות פעם אחת בכל אחת מחרוזות הקלט (כלומר, האותיות שופיעות בשתי המחרוזות). האותיות יודפסו לפי סדר האלף-בית, אותן בכל שורה.

למשל הקראיה

```
findLetters( "computer science" , "the child is smiling " )
```

תדפיס

c
e
i
m
n
s
t

שימוש לב ליעילות הפונקציה

עמוד 10 מתוך 15

מספר מהברת:

ת'ז:

```
void findLetters(char str1[], char str2[])
```

עמוד 11 מתוך 15

מספר מחברת:

ת'ז:

שאלה 5 א (10 נקודות):

כתבו מימוש **רקורסיבי** של הפונקציה:

```
int sumMod3(int num);
```

הfonקציה מקבלת כקלט `num` מספר שלם חיובי ומחזירה את סכום כל הספרות שמתחלקות ב-3.

דוגמאות:

הקריאה `sumMod3(23462)` תחזיר 9 כי $3+6=9$.

הקריאה `(sumMod3(28))` כי אין במספר ספרות המתחלקות ב-3.

הקריאה `(sumMod3(9))` תחזיר 9.

```
int sumMod3(int num)
```

שאלה 5 ב (10 נקודות):

כתבו מימוש **רקורסיבי** של הפונקציה

```
bool isPrefix(char str1[], char str2[]);
```

הfonקציה מקבלת כקלט שתי מחרוזות str1 ו-str2, ובודקת ומוחזירה אם ורק אם str2 היא רישא של str1, כלומר, מופיעיה בהתחלה str1.

דוגמאות

הקריאה `isPrefix("Daniel", "Dan")` נמצאת בתחילת `Daniel`.

הקריאה `isPrefix("Daniel is nice", "Daniel i")` תחזיר true.

הקריאה `isPrefix("Daniel is nice", "")` תחזיר true (המחרוזת ריקה).

הקריאה `isPrefix("", "")` תחזיר true (שתי המחרוזות ריקות).

אבל

הקריאה `isPrefix("Daniel is nice", "F")` תחזיר false.

וגם הקריאה `isPrefix("Daniel is nice", "Dan is")` תחזיר false.

עמוד 15 מתוך 15

מספר מהברת:

ת'ז:

```
bool isPrefix(char str1[], char str2[])
```