

סמסטר א' תשע"א מועד ב
 תאריך: 14.2.2011
 שעה: 14:00
 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 כללי תכונות מודולארי. הגדרו קבועים היכן שצורך וכתבו פונקציות עזר.
- יש לשחרר את כל שטחי הזיכרון שהוקטו 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 (ambil למש) בכל הפונקציות הנтונות בעמוד 2 של מבחן זה.
- אין צורך לטעד את הפונקציות.

- יש לכתוב ת"ז ומספר מחברת בראש כל עמוד. עמוד ללא ת"ז בראשו לא יבדק.
- יש לענות על כל שאלה במקום שמיועד לה בגוף השאלה. המחברות הן טיוותה בלבד ולא תיבדקנה.
- שימו לב כי הבחינה כוללת 15 עמודים, כולל עמוד זה.

בהצלחה!

רשימת פונקציות ספריה מותרונות לשימוש:

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

שאלה 1 (20 נקודות):

הנicho כי מוגדר הקבוע הבא:

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

כתבו את הפונקציה:

```
int sumFrame(int mat[][][MAX_SIZE], int matSize,
             int frameSize);
```

הפונקציה מקבלת כקלט את `Mat`, מטריצה ריבועית (מספר שורות = מספר עמודות) של מספרים שלמים. הפונקציה מקבלת גם את `matSize`, גודלה הלוגי של `Mat`, שמצוין כמה שורות יש במטריצה וכן גם כמה עמודות במטריצה $(matSize \leq MAX_SIZE)$.

הפונקציה מחשבת ומחזירה את סכום המספרים במסגרת בגודל `frameSize` (שלם חיובי). בחישוב סכום המספרים בתחום יש לסקם את המספרים הנמצאים

ב:

- `השורות הראשונות frameSize`
- `השורות האחרונות frameSize`
- `העמודות הימניות frameSize`
- `העמודות השמאליות frameSize`

שימוש לב שיש לסקם בדיק פעם אחת כל מספר, למשל לסקם רק פעם אחת את המספר המופיע בשורה הראשונה בעמודה הימנית.

הנicho ש- $frameSize < 0.5 * matSize$

דוגמה: עבור המטריצה `Mat` הבאה (שבורה $= 7$)

3	16	7	5	14	3	-2
-1	2	-3	-2	15	-7	0
14	2	-70	65	2	13	4
7	6	5	4	3	2	1
4	32	14	-21	3	-12	23
2	13	14	3	-2	0	0
12	12	23	4	-3	-2	1

הקריאה `sumFrame(Mat, 7, 1)` מסומם את המספרים שבמשבצות הכהות (כל מספר מסומם בדיק פעם אחת)

3	16	7	5	14	3	-2
-1	2	-3	-2	15	-7	0
14	2	-70	65	2	13	4
7	6	5	4	3	2	1
4	32	14	-21	3	-12	23
2	13	14	3	-2	0	0
12	12	23	4	-3	-2	1

וتحזיר את הסכום :

$$\begin{aligned}
 & -2+3+14+5+7+16+3+0-1+4+14+1+7+23+4+0+2+1-2-3+4+23+12+12 \\
 & =147
 \end{aligned}$$

ואילו הקריאה `sumFrame(Mat, 7, 3)` מסומם את המספרים שבמשבצות הכהות (כל מספר מסומם בדיק פעם אחד)

3	16	7	5	14	3	-2
-1	2	-3	-2	15	-7	0
14	2	-70	65	2	13	4
7	6	5	4	3	2	1
4	32	14	-21	3	-12	23
2	13	14	3	-2	0	0
12	12	23	4	-3	-2	1

וتحזיר את הסכום 224.

```
int sumFrame(int Mat[][][MAX_SIZE], int matSize,
             int frameSize)
{
    int i,j ;
    int sum ;

    sum = 0 ;

    //first frameSize lines
    for (i = 0 ; i < frameSize ; i++)
        for (j = 0 ; j < matSize ; j++)
            sum += Mat[i][j] ;

    //middle (matSize - 2frameSize) lines
    for (i = frameSize ; i < matSize - frameSize ; i++)
    {
        for (j = 0 ; j < frameSize ; j++)
            sum += Mat[i][j] ;
        for (j = matSize - frameSize ; j < maSize ;
             j++)
            sum += Mat[i][j] ;
    }

    //last frameSize lines
    for (i = matSize - frameSize ; i < matSize ; i++)
        for (j = 0 ; j < matSize ; j++)
            sum += Mat[i][j] ;

    return (sum) ;
}
```

שאלה 2 (20 נקודות): **כתבו את הפונקציה**

```
bool oneZone(int arr[], int size);
```

הfonקציה מקבלת כקלט מערך `arr` של מספרים שלמים אי-שליליים, הנמצאים בתחום $1 \dots 3^{size-1}$ ו-`size` (שם חיובי).

הfonקציה מחזירה `true` אם ורק אם כל אחד מהמספרים במערך מופיע בו בדיקות אחת (שים לב שלא יתכו שכל המספרים בתחום מופיעים במערך).

דוגמאות

הקריאה `oneZone([13,12,4,5,1,2],6)` תחזיר את הערך `true`, כי כל מספר מופיע בבדיקה פעמיות אחת.

הקריאה `oneZone([1,13,4,5,13,2],6)` תחזיר את הערך `false`, כי `13` מופיע פעמיים.

```
bool one2one(int arr[], int size)
{
    int i;
    bool res ;
    bool *foundarr ;

    foundarr = new bool[3*size] ;
    for (i = 0 ; i < 3 * size ; i++)
        foundarr[i] = false ;

    res = true ;
    for (i = 0 ; (i < size) && res ; i++)
    {
        if ( ! foundarr[ arr[i] ] )
            foundarr[ arr[i] ] = true ;
        else // already found
            res = false ;
    }

    delete [] foundarr;
    return(res) ;
}
```

שאלה 3 א (12 נקודות):

כתבו מימוש **רקורסיבי** של הפונקציה:

```
bool notFar(int data[], int size, int dist);
```

הפונקציה מקבלת כקלט מערך `data` של מספרים שלמים, ואת גודלו `size` (שלם חיובי) ומספר שלם חיובי `dist`. הפונקציה מחזירה ערך `true` אם ורק אם המרחק בין כל שני מספרים סמוכים במערך (כלומר באינדקסים עוקבים) הוא לכל היותר `dist`.

הערה: המרחק בין שני ערכים מוגדר להיות הערך המוחלט של הפרשיהם.

דוגמאות:

הקריאה `.true notFar([-2,3,5,1,-4,-3],6,5)` תחזיר `false` כי המרחק בין 1 ל-5 גדול מדי.

```
bool notFar(int data[], int size, int dist)
{
    bool res ;

    if (size == 1)
        return( true) ;
    else
    {
        res = notFar(data, size -1 , dist) ;
        if (res)
        {
            return
                ( (data[size-2] - data[size-1] < dist)&&
                   (data[size-1] - data[size-2]< dist));
        }
        else
            return (false) ;
    }
}
```

שאלה 3 ב (13 נקודות):**כתבו מימוש רקורסיבי של הפונקציה**

```
int biggestLowPower(int x, int num);
```

הfonקציה מקבלת שני מספרים שלמים חיוביים x ו- num , ומחזירה את החזקה המכסימאלית של x (x^y , עבור y שלם אי-שלילי), הקטנה או שווה ל- num .

דוגמאות

הקריאה (2^8) 256 **תחזיר** `biggestLowPower(2, 256)`

הקריאה (2^7) 128 **תחזיר** `biggestLowPower(2, 200)`, (2^8 גדול מ-200).

הקריאה (12^1) 12 **תחזיר** `biggestLowPower(12, 20)`

הקריאה (12^0) 1 **תחזיר** `biggestLowPower(12, 2)`

```
int biggestLowPower(int x, int num)
{
    if (x > num)
        return(1) ;
    else
        return ( x* biggestLowPower(x, num/x) )
}
```

שאלה 4 (15 נקודות):**לרשוטכם עומדת הפונקציה:**

```
int strStr( char s1[], char s2[] ) ;
```

הפונקציה מקבלת כקלט שני מחרוזות s_1 ו- s_2 , כאשר s_2 אינה ריקה, וממחשת את המופיע הראשון של המחרוזת s_2 במחרוזת s_1 ומחזירה את מיקומו או -1 אם s_2 אינה מופיעה ב- s_1 .

דוגמאות

הקריאה `strStr("abcabcdedede", "abc")` תחזיר 0.

הקריאה `strStr("abcabcdedede", "de")` תחזיר 6.

הקריאה `strStr("abcabcdedede", "ef")` תחזיר 1- כי המחרוזת ef לא מופיעה במחרוזת הראשונה.

הקריאה `strStr("abcabcdedede", "dd")` תחזיר 1- כי המחרוזת dd לא מופיעה במחרוזת הראשונה.

הערה: אין צורך למש פונקציה זו!!!

כתבו את הפונקציה:

```
int lastStrStr(char s1[], char s2[]);
```

תוך שימוש בפונקציה `strStr` כפונקציה עוזר.

הפונקציה מקבלת כקלט שתי מחרוזות s_1 ו- s_2 , כאשר s_2 אינה ריקה, וממחשת את המופיע האחרון של המחרוזת s_2 במחרוזת s_1 ומחזירה את מיקומו או -1 אם s_2 אינה מופיעה ב- s_1 .

דוגמאות

הקריאה `lastStrStr("abcabcdedede", "abc")` תחזיר 3.

הקריאה `lastStrStr("abcabcdedede", "de")` תחזיר 10.

הקריאה `lastStrStr("abcabcdedede", "ef")` תחזיר -1.

הקריאה `lastStrStr("abcabcdedede", "dd")` תחזיר -1.

```
int lastStrStr(char s1[], char s2[])
{
    int place ;
    int currPlace ;

    place = -1 ;
    currPlace = strStr(s1,s2) ;

    while(currPlace != -1)
    {
        place += (currplace +1) ;
        // Adding one:
        //in the first time to overcome the
        //initial value -1, in the other times
        //to overcome the fact we started from
        //place+1
        currplace = strStr(s1 + place + 1 , s2) ;
    }

    return(place) ;
}
```

שאלה 5 (20 נקודות):

נתונים הקבועים הבאים:

```
const int MAX_NAME_LEN = 100;
const int MAX_NUM_OF_CHILDREN = 20;
```

נתונה ההגדרה הבאה לייצוג משפחה:

```
struct family{
    char name[MAX_NAME_LEN];           // שדה המכיל את שם המשפחה
    int numChildren;                  // שדה המכיל את מספר הילדים
    int ages[MAX_NUM_OF_CHILDREN];    // מערך של גילאי הילדים
};

typedef struct family Family;
```

כתבו את הפונקציה הבאה:

```
int numChildren(Family fams[], int size, char ch,
                int minAge);
```

הפונקציה מקבלת כקלט מערך `fams` של משפחות ואת גודלו `size` (שלם חיובי), כמו כן הפונקציה מקבלת כקלט אות `ch`, ושלם `.minAge`.
 הפונקציה מחזירה את המספר הכלול של הילדים שגילם לפחות `minAge`, ושם משפחתם מתחילה באות `ch`.

```
int numChildren(Family fams[], int size, char ch,
                int minAge)
{
    int i ;
    int count ;

    count = 0 ;

    for (i = 0 ; i < size ; i++)
    {
        if (fams[i].name[0] == ch)
        {
            for (j = 0; j < fams[i].numChildren ; j++)
                if (fams[i].ages[j] >= minAge)
                    count += fams[i].ages[j] ;
        }
    }

    return(count) ;
}
```