

```

void f1();
void f2(int x);
void f3(int x);

```

- קריטריון -

257

```

void f3(int x)
{
    cout << f3 << "\n";
    return 10 * x;
}
void f2(int x)
{
    int y;
    cout << f2 - 1 << "\n";
    y = f3(x);
}
void f1()
{
    cout << f1 - 1 << "\n";
    f2(3);
    cout << f1 - 2 << "\n";
}
main()
{
    cout << main - 1 << "\n";
    f1();
    cout << main - 2 << "\n";
}

```

main-1
f1-1
f2-1
30
f1-2
main-2

## תרגול פונקציות

1. מה תהיה כותרת הפונקציה בשאלות הבאות: (שימי לב לפרמטרים ולסוג הערך המוחזר)

- א. פונקציה המקבלת אורך ורוחב של מלבן ומשרטת מלבן מכוכבים בגודל זה.
- ב. פונקציה המקבלת כקלט מחיר נסיעה ומחזירה בתשובה מחיר לאחר העלאה של 15%.
- ג. פונקציה המציירת מסך פתיחה של משחק.
- ד. פונקציה המדפיסה תפריט על המסך ומחזירה את בחירת המשתמש.
- ה. פונקציה המקבלת מספר תעודת זהות ומחזירה את ספרת הביקורת שלו.
- ו. פונקציה המקבלת מערך של 30 ציונים ומחזירה את הממוצע המדויק שלהם.
- ז. פונקציה המקבלת מערך מספרים ומאפסת את כל המספרים השליליים במערך.

2. לפניך כותרות של פונקציות:

1. void pticha( )
2. int bdika(int mis1, int mis2)
3. int kamut(int muzmanim)
4. void hadpasa( int godel)
5. float derech(float zman, float meirut)
6. int godel( )
7. long binary (int mis)
8. int kayam (int mis[], int n)
9. void ziun(int godel)

עני על השאלות הבאות:

- א. מהם הפונקציות שאין להם ערך מוחזר?
- ב. מהם הפונקציות שאינן מקבלות פרמטרים
- ג. לאילו פונקציות יותר מפרמטר אחד?
- ד. אילו פונקציות מקבלות ערכים וגם מחזירות ערכים?
- ה. האם יש קשר בין סוג הפרמטרים לסוג הערך המוחזר?
- ו. כתבי פעולת return לפונקציה 9 אם מטרתה להוסיף לציון המקורי 5 נקודות אם הוא מעל 80.
- ז. כתבי את פונקציה 3 אם מטרתה לחשב את מספר השולחנות הדרוש לסעודה לפי מספר הזמזומנים, אם סביב כל שלחן יושבים 6 סועדים.
- ח. כתבי 2 הפעולות לכל פונקציה. (עם שינוי בפרמטרים ובשימוש בערך המוחזר)

— 10/10/18

void malben(int width, int height)

10 (1)

float mechi(float a)

2

void open()

1

int menu()

3

int tz(long id)

7

1, 4, 9

10 (2)

1, 6

2

2, 5, 8

2

2, 3, 5, 7, 9

3

15

7

int ziun(int godel)

1

{ if(godel > 80)

return godel + 5;

}

int kamut(int muzmanin)

5

{ if(muzmanin % 6 == 0)

return muzmanin / 6;

return muzmanin + 1;

}

} return muzmanin / 6 + (muzmanin % 6);

7

## תרגול - פונקציות

1. כתבי פונקציה f1 המקבלת 2 מספרים ומחזירה את הגדול ביניהם.
2. כתבי פונקציה f2 המקבלת 3 מספרים ומחזירה את הגדול ביניהם.
3. שכתבי את הפונקציה f2 משאלה 2 כך שתשתמש ב-f1.
4. כתבי פונקציה f3 המקבלת 2 מספרים ומחזירה האם הראשון גדול מהשני.
5. כתבי פונקציה המקבלת 2 מספרים שלמים a ו-b, הפונקציה מדפיסה את המנה ושארית החלוקה של a ב-b. החישוב יתבצע ללא האופרטורים  $\%$  / (ע"י לולאה ואופרטור -).
6. כתבי תוכנית הקולטת מספר ומשרטטת משולש כוכביות בצורה הבאה:

לקלט 7

```
*  
* * *  
* * * * *  
* * * * * *  
0 0 0 0 0 0 0 0
```

התוכנית נעזרת בפונקציות הבאות:

- f1 לבדיקת המספר שנקלט: חיובי ואי זוגי.
  - f2 לשרטוט x רווחים, ה-x הוא פרמטר.
  - f3 לשרטוט x כוכביות, ה-x הוא פרמטר.
  - f4 לשרטוט המשולש: חישוב מספר הרווחים והכוכביות הדרושים, והפעלת f2 ו-f3.
- main להפעלת הפונקציות הדרושות.



→ 1.3/10 - 1/18/22

→ 6/2

int f1(int a, int b)

{ if (a > b)

return a;

return b;

}

if else  
↑ ↑  
return a > b ? a : b;

(1)

void f2(int x, int y, int z)

{ if (x > y && x > z

return x;

if (y > x && y > z)

return y;

return z;

void f3(int x, int y, int z)

{ int m = f1(x, y); return f1(z, f1(x, y));

return f1(m, z);

}

int f3(int a, int b)

{ if (a > b)

return a;

return b;

return a > b;

}

int main()

{ int a, b;

cout << "press 2 numbers";

cin >> a >> b;

cout << f3(a, b);

(2)

(3)

(4)

- पर -

```

void f4(int num)
{
    int i;
    for(i=1; i<=num; i++)
        f2(num-i/2);
    f3(i);
    cout<<"\n";
}

```

```

int f1(int num)
{
    return (num>0?f1(num/2);
}

```

```

void f2(int x)
{

```

```

    while(x--)
        cout<<" ";
}

```

```

void f3(int x)
{

```

```

    while(x--)
        cout<<"*";
}

```

```

void main()
{
    int num;
    cout<<" ";
    cin>>num;
    if(f1(num))
        f4(num);
}

```

- תרגיל -

1. כתבי פונקציה לקליטת מספר מהמשתמש בטווח מסוים, הפונקציה מחייבת להקיש מספר נכון! ומחזירה את המספר.
2. כתבי פונקציה להדפסת  $\sum$  (סיגמה) של מספר. (קלט - n פלט - הסכום  $1+2+3+\dots+n$ )
3. כתבי פונקציה המקבלת 2 מספרים ומדפיסה את כל המספרים השלמים שביניהם.
4. כתבי פונקציה בוליאנית המקבלת מספר ובודקת האם הוא מתחלקת ב-5 ללא שארית.
5. כתבי פונקציה המקבלת 2 מספרים ומחזירה את המחלק המשותף הגדול ביותר.
6. כתבי פונקציה המקבלת מספר שלם וספרה ומחזירה את מספר הפעמים שספרה זו מופיעה במספר. אם הקלט לא חוקי מחזירה ערך -1.
7. התכונה הבאה ידועה כנכונה עבור מספרים שלמים בתחום 1 עד 100. מכל מספר ניתן לפרוש סידרה כמפורט להלן:

$$X_1 = X$$

$$X_{n+1} = X_n / 2 \quad \text{אם } X_n \text{ זוגי}$$

$$\text{אם } X_n \text{ אי זוגי } 3X_n + 1$$

10, 5, 16, 8, 4, 2, 1  
1, 4, 2, 1

הסדרה תסתיים בספרה 1  
לדוגמא הסדרה הנפרשת מהמספר 10:  
מהמספר 1:

כתבי פונקציה המקבלת מספר. הפונקציה בודקת אם המספר בתחום 1 - 100, אם לא מחזירה ערך אפס. הפונקציה תדפיס את הסדרה שנפרשה מהמספר ותחזיר האיבר הגדול ביותר בסדרה שנפרשה.

8. מה הוא פלט ההדפסה של התכנית הבאה:

```
int func1(int x, in y)
{
    x+=2;
    y+=2;
    cout<<y<<" = y "<<x<<" = x "<<" הם הערכים ה- func1 הפונקציה הערכים הם:
    return(x*y);
}

void main()
{
    int x, y, i;
    x=2;
    y=4;
    for(i=0;i<3;i++)
    {
        x=func1(x,y);
        cout<<y<<" = y "<<x<<" = x "<<" for - של לולאת ה-
    }
}
```

9. כתבי פונקציה המקבלת מערך ובו ערכי מניה בכל אחד מהחודשים בשנה הקודמת, הפונקציה תחזיר את מספר החודש בו היה השינוי הגדול ביותר בערכה של המניה.
10. כתבי פונקציה המקבלת כפרמטרים - מערך מספרים וגודל המערך, ומחליפה כל איבר בסכום האיברים הקודמים לו.



program

302

```
int fun()
{
    int x;
    cout << "press a number";
    cin >> x;
    while (x < 0 || x > 100)
    {
        cout << "in 1st part";
        cin >> x;
    }
    return x;
}

void main
{
    cout << fun();
}
```

(1)

```
int fun()
{
    int x, sum;
    cout << "press a number";
    cin >> x;
    while (x > 0)
    {
        sum += x;
        x--;
    }
    return sum;
}

void main
{
    cout << fun();
}
```

(2)

```
int fun(int x, int y)
{
    if (x > y)
    {
        while (x > y)
        {
            cout << x;
            x--;
        }
    }
    else
    {
        while (y > x)
        {
            cout << y;
            y--;
        }
    }
}

void main
{
    cout << "press two numbers";
    int x, y;
    fun(x, y);
}
```

(3)



-person-

202

int fun()

{

int x;

cout << "press a number" << endl;

cin >> x;

if (x % 51 == 0)

return 1;

return 0;

}

void main()

{

int fun(int x, int y)

{

5

- 725 -

```
int fun1(long x)
{
    int t;
    t = x % 10;
    return t;
}

int fun()
{
    long x, y, t, c;
    cout << "press a number and number";
    cin >> x >> y;
    s = fun1(x);
    if (fun1(y) == s)
        c = 1;
    fun1(x / 10);
    return c;
}

return -1;
```

```
void mark()
{
    cout << fun();
}

int fun(int x)
{
    int max = 0;
    if (x < 0 || x > 100)
        return 0;
    else
    {
        while (x != 1)
        {
            if (x % 2 == 0)
            {
                if (x > max)
                    max = x;
                x /= 2;
            }
            cout << x;
        }
        else
        {
            if (x > max)
                max = x;
            x = 3 * x + 1;
        }
        cout << x;
    }
    return max;
}

void mark()
{
    int x;
    cout << "press a number";
    cin >> x;
    cout << fun(x);
}
```

(c)

f

- 727 -

```
int fun(int x)
```

```
{  
    int max = 0;
```

```
    if (x > 100 || x < 0)
```

```
        return 0
```

```
    else
```

```
    do
```

```
    { if (x % 2 == 0)
```

```
        if (x > max)
```

```
            max = x;
```

```
            x = x / 2;
```

```
        cout << x;
```

```
    }  
    else
```

```
    { if (x % 2 != 0)
```

```
        if (x > max)
```

```
            max = x;
```

```
            x = x * 3 + 1;
```

```
        cout << x;
```

```
    }  
    while (x != 1);
```

```
    - cout << x;
```

```
    return max;
```

```
void main()
```

```
{  
    int num;
```

```
    cout << "press a number";
```

```
    cin >> num;
```

```
    cout << fun(int num);  
}
```

70111-

902

```
int fun( int arr[], int leng)
{
    int b, i, max = 0;
    for(i=0; i < leng; i++)
    {
        if(arr[i+1] - abs(arr[i]) > max;
        {
            max = arr[i+1] - abs(arr[i]);
            b = i;
        }
    }
    return b;
}
```

void main

int arr[12];

cout << fun(arr, 12)

void fun( int arr[], int leng)

{
 int i, sum = 0;

for(i=0; i < leng; i++)

{
 arr[i] += sum;

sum += arr[i];

for(i=0; i < leng; i++)

{
 cout << arr[i];

void main

int i, arr[5];

cout << "press 5 numbers";

for(i=0; i < 5; i++)

cin >> arr[i];

fun(arr, 5);



- סיכום

207

המטרה היא למצוא את האינדקס של האיבר הראשון ששווה ל-x במערך arr. אם לא נמצא, להחזיר -1.

```
int fun(int arr, int len, int x)
{
    int i;
    for(i=0; i<len; i++)
        if(arr[i]==x)
            return i;
    return -1;
}
```

void main

```
int arr[10], x=5;
cout << fun(arr, 10, x);
```

## מספרים אקראיים

### הפונקציות rand ו-srand

```
srand((unsigned)time(NULL));
```

```
rand();
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

יש להכליל את הקבצים:

rand - הפונקציה מחזירה מספר אקראי.

srand - הפונקציה טוענת ערך אקראי התחלתי משעון המערכת. גורמת שבכל הרצה של התכנית, הפונקציה rand תגריל מספרים אקראיים בסדר שונה.

כדי להגדיר את הטווח אנחנו משתמשים באופרטור מודולו (%) אחרי ה rand יבוא % ואחריו הטווח המקסימלי.

לדוגמא:  $num = rand() \% 10$  מספר חד ספרתי

כדי להגריל מספר בטווח x (המספר הנמוך) עד y (המספר הגבוה)  $num = rand() \% (y - x + 1) + x$

לדוגמא:  $num = rand() \% 10 + 1$  מספרים בין 1 ל-10 כולל

$num = rand() \% 5 + 10$  המחשב יגריל מספר מבין המספרים: 10, 11, 12, 13, 14

### תרגול מספרים אקראיים

1. ישמי תכנית שהפלט שלה הוא 10 מספרים שונים בתחום:
  - a. 1-10
  - b. 5-250
  - c. מספרים זוגיים בין 1-20
  - d. שני מספרים שסכומם קטן מ-20
  - e. שני מספרים חד ספרתיים שסכומם גדול מ-10
2. כתבי תכנית שתציג 10 תרגילים שונים במספרים דו ספרתיים (הפעולות החשבוניות גם אקראיות) התוכנית תקלוט תשובות לתרגילים הפלט: ציון לכל תרגיל וציון סופי.
3. צרי שומר מסך שבכל רבע שניה מציב \* במקום אקראי במסך, יש לדאוג שלא יהיו יותר מ-50 כוכביות על המסך. פעילות שומר המסך תפסק בעת הקשת מקש על המקלדת.
4. כתבי תכנית מחשב המשחקת בול פגיעה כנגד המשתמש. המחשב בוחר מספר בן 5 ספרות, יש לדאוג שספרה לא תחזור על עצמה במספר הניחוש. על המשתמש לנחש את המספר ע"פ סדר הספרות. התכנית תכלול פונקציה המקבלת 2 מערכים מערך המספר ומערך הניחוש. הפונקציה מדפיסה את מס' הבולים והפגיעות ומחזירה את מס' הבולים. למשתמש מוקצבים 10 ניסיונות. אם הוא לא הצליח - התוכנית מדפיסה הודעה ומציגה את המספר הנבחר. אם המשחק נחש, התכנית מדפיסה הודעה מתאימה ואת מספר הניחושים.

```
1 // ConsoleApplication1.cpp : Defines the entry point for the console application.
2 //
3 //2 תרגול מספרים אקראיים תרגיל
4 #include "stdafx.h"
5 #include <conio.h>
6 #include <iostream>
7 #include <stdlib.h>
8 #include <time.h>
9 using namespace std;
10 //void main()
11 //{
12 // int x, y, z, i, t, mark=0, sum = 0;
13 // srand((unsigned)time(NULL));
14 // for (i = 0; i < 10; i++)
15 // {
16 //     y = rand() % 90 + 10;
17 //     z = rand() % 90 + 10;
18 //     x = rand() % 4;
19 //     switch (x)
20 //     {
21 //     case 0:y + z;
22 //         cout << y << "+" << z << "= ";
23 //         cin >> x;
24 //         t = y + z;
25 //         break;
26 //     case 1:y - z;
27 //         cout << y << "-" << z << "= ";
28 //         cin >> x;
29 //         t = y - z;
30 //         break;
31 //     case 2:y * z;
32 //         cout << y << "*" << z << "= ";
33 //         cin >> x;
34 //         t = y * z;
35 //         break;
36 //     case 3:y / z;
37 //         cout << y << "/" << z << "= ";
38 //         cin >> x;
39 //         t = y / z;
40 //         break;
41 //     }
42 //     if (t == x)
43 //         mark = 10;
44 //     sum += mark;
45 //     cout << "\n the mark is:" << mark<<"\n";
46 //     mark = 0;
47 // }
48 // cout << "\n the end mark is:" << sum;
49 // _getch();
50 //}
51 void main() 4 /for
52 {
53     int num[5], i, help[10] = { 0 }, j, guess[5], k, count1 , count2 ;
54     srand((unsigned)time(NULL));
55     for (i = 0; i < 5; i++)
```

```
56     {
57         do
58         {
59             num[i] = rand() % 10;
60             help[num[i]]++;
61         } while (help[num[i]] > 1);
62     }
63     for (i = 0; i < 10; help[i++] = 0);
64     for (j = 0; j < 10; j++)
65     {
66         count1 = 0;
67         count2 = 0;
68         cout << "guess 5 numbers";
69         for (i = 0; i < 5; i++)
70         {
71             cin >> guess[i];
72             help[guess[i]]++;
73             while (help[guess[i]] > 1)
74             {
75                 cout << "press again";
76                 cin >> guess[i];
77                 help[guess[i]]++;
78             }
79         }
80         for (i = 0; i < 5; i++)
81         {
82             for (k = 0; k < 5 && guess[i] != num[k]; k++);
83             if (k < 5)
84             {
85                 if (k == i)
86                     count1++;
87                 else
88                     count2++;
89             }
90         }
91         cout << "\n bul: " << count1 << "pgiah: " << count2;
92         if (count1 == 5)
93         {
94             cout << "\n wow, " << j + 1 << "guess";
95             break;
96         }
97         for (i = 0; i < 10; help[i++] = 0);
98     }
99     if (j == 10)
100         cout << "\n soory,10 times has passed";
101     _getch();
102 }
103
104
```



שאלה מס' 3 (16 נק')

כתוב פונקציה אשר מקבלת כפרמטרים שני מערכים של מספרים שלמים בין 1 ל-99 באורך 20.

הפונקציה תבצע את הפעולות הבאות:

1. תבדוק את חוקיות המספרים שבמערכים. (1-99). קאמלסון טונקליג. לו ח'ת' לו גלגל בקר  
מקלל מלוק ומתפיה צין ס א, חח אוקי אחיו 1  
תמצא ותדפיס את המספרים אשר נמצאים רק באחד מהמערכים.
  2. תמצא ותדפיס את המספר שמופיע הכי הרבה במערך הראשון וכן תדפיס כמה פעמים הוא מופיע.
  3. תמצא ותדפיס את המספר שמופיע הכי הרבה במערך השני וכן תדפיס כמה פעמים הוא מופיע.
  4. תמצא ותדפיס את המספר שמופיע בשני המערכים ביחד הכי הרבה פעמים וכן תדפיס כמה פעמים הוא מופיע.
- דוגמא:  $\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \end{matrix}$

$$A = 1,2,2,3,4,6,2,7,8,9,9,9,9,2,3,4,6,2,9$$

$$B = 2,2,2,3,4,5,7,8,9,7,12,23,4,5,5,5,5,4,4,7$$

המספרים שנמצאים רק באחד מהמערכים:

1,5,6,12,23

המספר שמופיע הכי הרבה במערך הראשון הוא: 9. מופיע 6 פעמים.

המספר שמופיע הכי הרבה במערך השני הוא: 5. מופיע 5 פעמים.

המספר שמופיע בשני המערכים ומופיע הכי הרבה הוא: 2. מופיע 8 פעמים.

```

1 // ConsoleApplication1.cpp : Defines the entry point for the console
  application.
2 //
3
4 #include "stdafx.h"
5 #include <conio.h>
6 #include <iostream>
7 using namespace std;
8 int fun1(int a[])
9 {
10     int i;
11     for (i = 0; i < 20; i++)
12     {
13         if (a[i] < 1 || a[i]>99)
14             return 0;
15     }
16     return 1;
17 }
18 void fun2(int a[], int b[])
19 {
20     int i;
21     for (i = 0; i < 20; i++)
22     {
23         a[b[i]]++;
24     }
25 }
26 void fun3(int a[], int b[])
27 {
28     int i;
29     for (i = 0; i < 100; i++)
30     {
31         if ((a[i] == 0 && b[i] > 0) || (a[i] > 0 && b[i] == 0))
32             cout << i;
33     }
34 }
35 void fun4(int a[])
36 {
37     int i, max = 0, numax;
38     for (i = 0; i < 100; i++)
39     {
40         if (a[i] > max)
41         {
42             max = a[i];
43             numax = i;
44         }
45     }
46     cout << "the number:" << numax << "is the max,is" << max << "times";
47     return numax;
48 }
49 void fun5(int a[], int b[])
50 {
51     int i, max = 0, numax;
52     for (i = 0; i < 100; i++)
53     {
54         if (a[i] + b[i] > max)
55         {
56             max = a[i] + b[i];

```

```

56     numax = i;
57     }
58     }
59     cout << "the number:" << numax << "is the max,is" << max << "times";
60     for (i = 0; i < 100; i++)
61     {
62         if (a[i] + b[i] == max&& i != numax)
63         {
64             cout << i; - 111'071 111 100
65         }
66     }
67 }
68 void main()
69 {
70     int a[20] = { 1,2,2,3,4,6,2,7,8,9,9,9,9,2,3,4,6,2,9}, b[20] =
        { 2,2,2,3,4,5,7,8,9,7,12,23,4,5,5,5,5,4,4,7}, helpa[100] = { 0 }, helpb
        [100] = { 0 };
71     fun1(a, 20);
72     fun1(b, 20);
73
74     (if[(!fun1(a)) || (!fun1(b))])
75     {
76         break;
77     }
78     else
79     {
80         if(!fun1(a) || !fun1(b))
81         {
82             fun2(helpa);
83             fun2(helpb);
84             fun3(helpa, helpb);
85             cout << fun4(helpa) << fun4(helpb);
86             fun5(helpa, helpb);
87         }
88     }
89     getch();
90 }

```

## חלק א' (40 נקודות)

ענו על שתיים מבין השאלות 1-3 (לכל שאלה – 20 נקודות).

### שאלה 1

(5 נק') א. כתבו פונקציה בשפת C אשר תוגדר באופן הבא:

```
int func1(int vec[], int num, int n)
```

הפונקציה מקבלת מערך של שלמים חיוביים vec בגודל n ומחזירה את המיקום של המספר num במערך. אם הוא לא קיים, תחזיר -1. אם המספר נמצא יותר מפעם אחת, תחזיר את המקום הראשון שנמצא.  
כאמור

(5 נק') ב. כתבו פונקציה בשפת C אשר תוגדר באופן הבא:

```
void func2(int vec[], int index, int n)
```

הפונקציה מקבלת מערך של שלמים חיוביים vec בגודל n, ופרמטר שלם index, שמייצג אינדקס במערך. הפונקציה תבדוק את חוקיות ה-index. אם הוא תקין, תמחק את האיבר שבמיקום index ותצמצם את המערך. אחרי הצמצום הפונקציה תציב את הערך -1 בסוף המערך.

(10 נק') ג. יש להשתמש בפונקציות שכתבתם בסעיפים הקודמים. כתבו פונקציה בשפת C אשר תוגדר באופן הבא:

```
void func3(int vec[], int n)
```

הפונקציה מקבלת מערך של שלמים חיוביים vec בגודל n ומוחקת איברים כפולים במערך. הפונקציה אינה משנה את גודל המערך. בסוף המערך המצומצם יוצבו הערכים -1 עד הגודל המקורי של המערך (n). הפונקציה תבדוק את חוקיות הקלט.

דוגמה:

נתון המערך vec הבא:

vec = 1, 2, 3, 4, 5, 4, 3, 6, 7, 8, 2, 11, 9

אחרי ביצוע func1:

func1(vec, 5, 13)

תחזיר: 4 שהוא המיקום שבו קיימת הספירה 5.

עבור

func1(vec, 13, 13)



תחזיר ו- מכיוון שאין איבר 13 במערך.

func2(vec,5,13)

אחרי ביצוע הפונקציה func2 הערכים במערך vec יהיו:

vec = 1, 2, 3, 4, 5, 3, 6, 7, 8, 2, 11, 9, -1

צומצם האיבר החמישי (שהיה 4). בסוף המערך הוצב המספר -1

אחרי ביצוע הפונקציה func3(vec,13)

vec = 1, 2, 3, 4, 5, 6, 7, 8, 11, 9, -1, -1, -1

## (שאלה 2)

כתבו תכנית בשפת C אשר תוגדר באופן הבא:

```
void func(char str[], int nums[])
```

הפונקציה מקבלת כפרמטרים מחרוזת של תווים, שכוללת אותיות ומספרים בלבד, ללא רווחים וסימנים אחרים. בנוסף הפונקציה מקבלת כפרמטר מערך של מספרים שלמים חיוביים שסופו מסתיים ב'-1.

5 נק') א. הפונקציה תבדוק את תקינות הקלט, כלומר שבמחרוזת יש רק אותיות ומספרים, וכן שבמערך ישנם מספרים חיוביים ושסופו -1. אם הקלט חוקי, יש לעבור לביצוע סעיף ב'. אם לא, יש להדפיס הודעה מתאימה ולסיים את הריצה.

15 נק') ב. המספרים שבמערך מציינים את אורך המילים שבמחרוזת, אשר אינם מופרדים ברווחים. על הפונקציה לבדוק את תקינות הקלט, כלומר שאכן המספרים שבמערך מייצגים את מספר האותיות שבמחרוזת. אם הקלט אינו תקין, יש להדפיס הודעה. אם הקלט תקין, הפונקציה תדפיס את המחרוזת כולל הרווחים בין המילים.  
דוגמה:

עבור המחרוזת :

hellothisisCexamsummer 2011

והמערך:

5, 4, 2, 1, 4, 6, 4, -1

הפונקציה תדפיס:

hello this is C exam summer 2011

1 line

25P

```
int func1(int vec[], int num, int n)
```

```
{ int a = -1;
```

```
for(i=0; i<n; i++)
```

```
{ if(vec[i] == num)
```

```
    a = i;
```

```
}
```

```
return a;
```

```
}
```

```
int func2(int vec[], int index, int n)
```

```
{ int i;
```

```
if(index > n)
```

```
return 0;
```

```
else for(i=index; i<n-1; i++)
```

```
{ vec[i] = vec[i+1];
```

```
}
```

```
vec[n] = -1;
```

```
void func3(int vec[], int n)
```

```
{ int i, a, tmp;
```

```
for(i=0; i<n; i++)
```

```
{ a = i;
```

```
while(a > 0)
```

```
{ if(vec[i] == -vec[i-1] || vec[i] == -1)
```

```
{ vec[i] = vec[i-1];
```

```
vec[i+1] = -1;
```

```
a--;
```

```
}
```

```
for(i=0; i<n; i++)
```

```
{ if(vec[i] == -1)
```

```
{ tmp = -1;
```

```
vec[i] = vec[i+1];
```

```
vec[i+1] = tmp;
```

```
}
```

```
}
```

```
void main()
```

```
{ int vec[13] = {1, 2, 3, 4, 5, 4, 3, 6, 7, 8, 2, 11, 9};
```

```
cout << func1(vec, 5, 13);
```

```
func2(vec, 5, 13);
```

```
func3(vec, 13);
```

```
getchar();
```

## פונקציות מתמטיות:

**abs(n)** – ערך מוחלט.

מקבל ערך: מספר שלם, התשובה: מספר שלם  
שייך לספריה - `stdlib.h`

**sqrt(n)** – שורש רבועי

מקבל מספר מסוג עשרוני, התשובה עשרונית.  
N – חייב להיות חיובי, לא ניתן להוציא שורש ממספר שלילי.  
שייך לספריה: `math.h`

**pow(x,y)** – העלאה בחזקה

חישוב הערך של X בחזקת Y.  
מקבל מספר עשרוני, התשובה: מספר עשרוני.  
אין להעלות 0 בחזקה שלילית.  
שייך לספריה: `math.h`

**pow10(n)** – חזקות של 10.

חישוב הערך של 10 בחזקת N.  
מקבל מספר עשרוני, התשובה: מספר עשרוני.  
שייך לספריה: `math.h`

**floor(n)** – עגול מספר עשרוני כלפי מטה

מקבל מספר עשרוני, התשובה: מספר עשרוני.  
שייך לספריה: `math.h`

**ceil(n)** – עגול מספר עשרוני כלפי מעלה.

מקבל מספר עשרוני, התשובה: מספר עשרוני.  
שייך לספריה: `math.h`

**round(n)** – עגול מספר עשרוני בצורה המתמטית הרגילה.

מקבל מספר עשרוני, התשובה: מספר עשרוני.  
שייך לספריה `math.h`

**modf(n,&x)** – מפריד מספר עשרוני ל-2 החלק העשרוני והחלק השלם.

מקבל מספר עשרוני, התשובה: מספר עשרוני שהוא החלק העשרוני.  
מציב במשתנה X מסוג double את החלק השלם.

**min(x,y)** – מציאת המספר הקטן מבין 2 מספרים.

מקבל שני מספרים שלמים ומחזיר מספר שלם.  
שייך לספריה - `math.h`

**max(x,y);** – מציאת המספר הגדול מבין 2 מספרים.

מקבל 2 מספרים שלמים ומחזיר מספר שלם.  
שייך לספריה - `math.h`