

מדעי המחשב - 2 יחידות לימוד
פתרון בחינת הבטחות

פרק א

שאלה 1:

sum	k	k אי-זוגי?	פלט
0			18
2	1	כן	
	2	לא	
8	3	כן	
	4	לא	
18	5	כן	
	6	לא	
	7		

הערה: אין חובת רישום עמודת התנאי. עמודה זו מסייעת בביצוע המעקב.

שאלה 2:

טענת כניסה: 2 מספרים שלמים.
טענת יציאה: הערך המוחלט של ההפרש שבין שני המספרים.

אפשרות א:	אפשרות ב:
<pre>int sub (int a, int b) { return abs (a - b); }</pre> <p>משפט הזימון: $x = \text{sub}(5, 5);$</p>	<pre>int sub (int a, int b) { if (a > b) return a - b; else return b - a; }</pre> <p>משפט הזימון: $x = \text{sub}(5, 5);$</p>

הערה: משפט הזימון יקבל כל שני מספרים זהים, או פעמיים אותו משתנה: $x = \text{sub}(k, k);$ בהנחה ש- k מוגדר ומאותחל.

שאלה 3:

א. הוראת קלט למחרוזת: `gets (tel);` או `scanf("%s", tel);`
 ב. אחת משתי האפשרויות הבאות:

II	I
<pre>strcpy (st1, "abc"); strcat (st1, tel);</pre>	<pre>string_type st1 = ""; strcat (st1, "abc"); strcat (st1, tel);</pre>

הערה: פעולת השרשור מוסיפה למחרוזת ולכן יש להתחיל ממחרוזת ריקה, או ממחרוזת אליה הושמה המחרוזת "abc" בהוראת `strcpy`.

ג.

II	I
<p>אורך המחרוזת הוא 5</p> <pre>printf ("%c \n", tel[3]); printf ("%c \n", tel[4]);</pre>	<p>אורך המחרוזת הוא k</p> <pre>printf ("%c \n", tel[k-2]); printf ("%c \n", tel[k-1]);</pre>

שאלה 4:

הערה: קטע תכנית אינו מחייב תיעוד (טענות כניסה ויציאה).

```
for (i = 0 ; i < 57 ; i++)
{
    d1 = a[i] % 10;           //--- ספרת האחדות
    d100 = a[i] / 100;      //--- ספרת המאות
    if (d100 == d1 * 2)
        printf ( " %d", a[i]);
}
```

שאלה 5:

count	num1	num2	num1 > num2	פלט
0				3
	15	8	כן	
1	8	7	כן	
2	7	3	כן	
3	3	4	לא	

א. טבלת מעקב:
 (אין חובת רישום של עמודת ההוראה, או עמודת התנאי).

ב. דוגמת קלט: $2, 3, \dots$
 (כל סדרת נתונים שבה המספר השני גדול מהראשון)

פרק ב

שאלה 6:

	0	1	2	3	4
a	8	7	6	5	2

א

flag	flag == 1	n	a[1] == n	k	k < 4	a[k] - 1 ≠ a[k+1]	פלט
1		8	כן	0	כן	לא	8 7
				1	כן	לא	7 6
				2	כן	לא	6 5
0				3	כן	כן	5 2
				4	לא		
	לא						הודעה ב

ב.

n = 8

	0	1	2	3	4
a	8	7	6	5	4

פלט "הודעה א"

ג.

flag ← 0

n = 9

	0	1	2	3	4
a	9	7	6	5	4

(i) במקום
אם במקום כלשהו במערך
אין מספר עוקב-יורד.

n = 9

	0	1	2	3	4
a	8	7	6	5	4

(ii) במקום
אם התא הראשון אינו 9
(שונה מ- 9).

שאלה 7:

```

//---          קלט: 30 שמות, לכל שם 3 ציונים.
//---          פלט: שם התלמיד וציונו הסופי. מספר התלמידים שקיבלו 100.
#include <stdio.h>

void main ()
{
    char name [20];          //--- שם התלמיד ---
    int grd1, grd2, grd3;    //--- שלושת הציונים ---
    float avg;              //--- ממוצע ---
    int i, count = 0;

    for (i = 0 ; i < 30 ; i++)
    {
        printf ("%d - שם התלמיד ה- → ", i);
        gets (name);
        printf (" 3 ציוניו → ");
        scanf ("%d %d %d", &grd1, &grd2, &grd3);

        avg = (float) (grd1 + grd2 + grd3) / 3;
        if (grd3 > 85)
        {
            avg = avg + 5;
            if (avg > 100)          ||          if (avg > 100 || avg == 100)
                avg = 100;        {
            if (avg == 100)        avg = 100;
                count ++;          count ++;
            }                      }
        printf ("%d = ציון , %s = שם \n", avg, name);
    }
    printf ("100 הציון את קיבלו תלמידים %d \n", count);
}

```

לפינה 8:

```

//--- קלט: פרטי השכלה וותק של 100 מועמדים. ---
//--- פלט: כמה מהמועמדים מתאימים לתפקיד בכיר. ---
#include <stdio.h>

int suits (int education, int years) ;

void main ()
{
    int education ;          //--- מספר שנות השכלה ---
    int years                //--- מספר שנות ותק ---
    int i, count = 0;
    for (i = 0 ; i < 100 ; i++)
    {
        printf (" → מספר שנות השכלה ומספר שנות ותק ");
        scanf ("%d %d", &education, &years);
        if (suits (education, years) == 2)
            count ++ ;
    }
    printf ("\n%d מתאימים לתפקיד בכיר", count);
}

//--- טענת כניסה: שנות השכלה ומספר שנות ותק. ---
//--- טענת יציאה: 0 - אם אינו מתאים, 1 - אם מתאים, 2 - אם מתאים לתפקיד בכיר. ---

int suits (int education, int years)
{
    int sel = education * years;
    if (sel < 8)
        return 0;
    else
        if (sel >= 8 && sel <= 15)
            return 1;
        else
            return 2;
}

```

פרק ג

לאלה 9:

```
//---      15x18 בגודל (מטריצה)
//---      פלט: הודעה האם המערך "פרחוני".
//---      מערך פרחוני הוא מערך שיש בו לפחות 5 "פרחים"
```

```
#include <stdio.h>
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
typedef int mat_type [15][18];
```

```
int is_flower (mat_type m, int i, int j);
```

```
void main ()
```

```
{
```

```
    mat_type m;
```

```
    int i, j, count = 0;
```

```
    for (i = 0 ; i < 15 ; i++) (*)
```

```
        for (j = 0 ; j < 18 ; j++)
```

```
            count = count + flower (m, i, j);
```

```
    if (count >= 5)
```

```
        printf (" מערך פרחוני \n");
```

```
    else
```

```
        printf (" מערך לא פרחוני \n");
```

```
}
```

```
//---      טענת כניסה: מטריצה m, שורה i ועמודה j
```

```
//---      טענת יציאה: "אמת" אם יש פרח ו-"שקר" אחרת.
```

```
//---      פרח = האיבר שבתא (i, j) שווה לסכום ארבעת האיברים שבארבע פינותיו.
```

```
int is_flower (mat_type m, int i, int j)
```

```
{
```

```
    int sum;
```

```
    if (i==0 || i==14 || j==0 || j==17)
```

```
        return FALSE;
```

```
    sum = m [i-1][j-1] + m [i-1][j+1] +
```

```
          m [i+1][j-1] + m [i+1][j+1];
```

```
    if (sum == m[i][j])
```

```
        return TRUE;
```

```
    else
```

```
        return FALSE ;
```

```
}
```

(*) בפונקציה flower נבדק ש-i ו-j אינם בקצוות. ללא בדיקה זו, תהיינה הלולאות:

```
for (i = 1 ; i < 14 ; i++)
    for (j = 1 ; j < 17 ; j++) ...
```

(*)

שאלה 10:

טבלת משתנים:

שם משתנה	טיפוס משתנה	תפקיד המשתנה
bead	char	צבע חרוז: 'G' = ירוק, 'Y' = צהוב, 'R' = אדום.
countG county countR	int	מונה חרוזים ירוקים, צהובים, אדומים <u>או</u>
colors	מערך בגודל 3 של שלמים	מערך מוני החרוזים
n	int	מספר החרוזים בשרשרת
valid	בוליאני 1/0	TRUE = 1 שרשרת תקינה, FALSE = 0 שרשרת לא תקינה.
count1	int	מספר מחרוזות
count2	int	מספר מחרוזות פגומות

תת-בעיות:

קלט-שרשרת -

טענת כניסה: מספר החרוזים.

טענת יציאה: מספר החרוזים הירוקים, צהובים ואדומים שנקלטו.

void **string_input** (int n , int * countG , int * countY , int * countR);

void **string_input** (int n , arr_type colors); או:

שרשרת-תקינה? (countG, county, countR)

טענת כניסה: 3 מונים של צבעי החרוזים שנקלטו.

טענת יציאה: "אמת" אם כל המונים שווים, ו-"שקר" אחרת.

int **valid_string** (int countG , int countY , int countR);

int **valid_string** (arr_type colors); או:

א. פתרון עם שלושה מונים.

```
//---          קלט: חרוזים בשרשרת.
//---          פלט: מספר השרשראות שיוצרו וכמה מתוכם פגומות.
#include <stdio.h>

void string_input (int n , int * countG , int * county , int * countR);
int valid_string (int countG , int countY , int countR);

void main ()
{
    int countG = 0, int countY = 0, int countR = 0;
    int count1 = 0 , count2 = 0, n;
    printf (" מספר החרוזים בשרשרת? ");
    scanf ("%d", &n);
    while (n != 0)
    {
        string_input (n , &countG , &county , &countR);
        count1 ++;
        if (! valid_string (countG , county , countR) ) count2 ++ ;
        printf (" מספר החרוזים בשרשרת? ");
        scanf ("%d", &n);
    }
    printf (" מחרוזות %d היו \n", count1);
    printf (" מחרוזות פגומות %d מתוכם \n", count2);
}

void string_input (int n , int * countG , int * county , int * countR)
{
    char bead ;          //--- צבע החרוז ---
    int i;
    for (i = 0 ; i < n ; i++)
    {
        printf (" צבע החרוז? ");
        bead = getchar();
        if (bead == 'G') *countG ++;
        else if (bead == 'Y') *county ++ ;
        else *countR ++;
    }
}

int valid_string (int countG , int countY , int countR)
{
    if (countG == countY && countG == countR)
        return TRUE;
    else return FALSE;
}
```

if (! valid_string (...)) ...
שקול להוראה:
if (valid_string (...) == FALSE) ...

ב. פתרון עם מערך מונים.

```

//--- קלט: חרוזים בשרשרת.
//--- פלט: מספר השרשראות שיוצרו וכמה מתוכם פגומות.
#include <stdio.h>

typedef int arr_type [3];

void string_input (int n , arr_type colors);
int valid_string (arr_type colors);

void main ()
{
    arr_type colors = {0};
    int count1 = 0 , count2 = 0, n;
    printf (" מספר החרוזים בשרשרת? ");
    scanf ("%d", &n);
    while (n != 0)
    {
        string_input (n, colors);
        count1 ++;
        if (! valid_string (colors) ) count2 ++ ;
        printf (" מספר החרוזים בשרשרת? ");
        scanf ("%d", &n);
    }
    printf (" מחרוזות %d היו \n", count1);
    printf (" מחרוזות פגומות %d מתוכם \n", count2);
}

void string_input (int n , arr_type colors)
{
    char bead ;           //--- צבע החרוז ---
    int i;
    for (i = 0 ; i < n ; i++)
    {
        printf (" צבע החרוז? ");
        bead = getchar();
        if (bead == 'G') colors[0] ++;
        else if (bead == 'Y') colors[1] ++'
        else colors[2] ++;
    }
}

int valid_string (arr_type colors)
{
    if (colors[0] == colors[1] && colors[0] == colors[2]) return TRUE;
    else return FALSE;
}

```

if (! **valid_string** (...)) ...
שקול להוראה:
if (**valid_string** (...) == FALSE) ...