

מדעי המחשב - 2 יחידות לימוד
פתרון בחינת הבטחות

פרק א

שאלה 1:

number	פלט
3	3
-7 7	7
-2 2	2
4	4
1	1
0	

א. טבלת מעקב לקלט: 0, 1, 4, -2, -7, 3 →

ב. האלגוריתם מדפיס את הערך המוחלט של המספרים שנקלטו.

שאלה 2:

טענת כניסה: הפונקציה מקבלת שני מספרים שלמים ל-a ול-b
טענת יציאה: הפונקציה מחזירה את הממוצע שלהם.

```
float mean (int a, int b)
{
    return (a+b)/2.0;
}
printf ("%f", mean (x,y) );
```

א. הקריאה לפונקציה:

שאלה 3:

א. להלן פתרונות שונים (התלמיד היה צריך להריץ פעם אחת בלבד)

a	b	I a<b	II a<100	I or II	פלט
150	50	שקר	שקר	שקר	שקר
70	50	שקר	אמת	אמת	אמת
150	250	אמת	שקר	אמת	אמת
50	70	אמת	אמת	אמת	אמת

ב. אם אין סוגריים הקובעים אחרת, אזי and קודם ל-or.

הקשר הלוגי האחרון לביצוע הוא or, ולכן מספיק שאחד הביטויים יהיה אמת כדי שערך כל הביטוי יהיה אמת.

$$x = -2, y = 5, z = 13$$

I	II	III	II and III	תוצאה
$z > x$	$x < 0$	$z - y > 9$		
אמת	אמת	שקר	שקר	אמת

שאלה 4:

התלמיד נדרש לכתוב קטע תכנית בלבד. בפתרון מוצגת תכנית מלאה.

דרך א: ניתן לכתוב קטע תכנית שיפסיק את החיפוש ברגע שמצא מחרוזת זהה. במקרה זה תהיה התשובה בסעיף ב: "ידיפס את מיקומה של המחרוזת הראשונה - במקום 2".

--- מילוי מערך של מחרוזות במחרוזות ---
 --- קלט מחרוזת והדפסת מיקומה במערך ---

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#define N 5 // should be 27
#define TRUE 1
#define FALSE 0

typedef char string_type [100];
typedef string_type arr_type [N];

void arr_kelet ( arr_type a );

void main ()
{
    arr_type a ;
    string_type st ;

    int found, i ;

    arr_kelet (a) ;
```

ניתן להגדיר את המערך כמערך של מצביעים למחרוזות:

השינויים בתכנית:

א. בשלב ההגדרה:

```
typedef char * string_type ;
typedef string_type arr_type [N] ;
```

ב. בפונקציית הקלט:

ג. בקלט המחרוזת הנוספת:

```

scanf ("%s", st); // gets (st);
found = FALSE;
i = 0;
while (! found && i < N)
{
    if (strcmp (st, a[i]) == 0)
        found = TRUE;
    i++;
}
printf ("%s was found at place: %d \n", st, i-1);
}
void arr_kelet ( arr_type a)
{
    int i;
    printf (" Type %d strings : ", N);
    for (i=0; i<N; i++)
        scanf ("%s", a[i]);
}

```

ג. קלט למחרוזת הנוספת:

```

string_type st;
char temp [100];
scanf ("%s", temp); // gets (temp);
st = strdup (temp);

```

ב. פונקציית הקלט:

```

void arr_kelet (arr_type a)
{
    int I;
    char temp [100];
    for (i=0; i<N; i++)
    {
        scanf ("%s", temp);
        a[i] = strdup (temp);
    }
}

```

דרך ב: ניתן לכתוב קטע תכנית שידפיס את מיקום המחרוזת בכל פעם שמוצא מחרוזת זהה, במקרה זה תהיה התשובה בסעיף ב: "יידפיס את מיקומן של כל המחרוזות הזהות - 2, 19".

```

void main ()
{
    :
    scanf ("%s", st); // gets (st);
    for (i=0; i<N; i++)
    {
        if (strcmp (st, a[i]) == 0)
            printf ("%s was found at place: %d \n", st, i);
    }
}

```

שאלה 5:

מ	i	פלט
0		3
1	0	
2	1	
3	2	

א. טבלת מעקב:

התכנית סופרת מדפיסה את מספר סימני ה-'x' באלכסון הראשי.

ב. אחרי השינוי יודפס מספר סימני ה-'x' בשורה 1 במטריצה.

פרק ב

שאלה 6:

```

//--- קלט גובה המבקרים בגן המשחקים ---
//--- קלט כמה גבוהים, כמה נמוכים ומהו הבונוס למפעילים ---
#include <stdio.h>
float get_height ();
void main ()
{
    float height ;
    int count_big = 0 , count_small = 0 ;
    height = get_height () ;
    while (height != 0)
    {
        if (height >= 1.70)
        {
            printf ("big \n") ;
            count_big ++ ;
        }
        else
        {
            printf ("small \n") ;
            count_small ++ ;
        }
        height = get_height () ;
    }
}

```

```

printf ("Number of BIG = %d \n", count_big) ;
printf ("Number of SMALL = %d \n", count_small) ;
printf ("Bonus for BIG = %d \n", count_big/15) ;
printf ("Bonus for SMALL = %d \n", count_small/20) ;
}

//--- מסננת קלט לבדיקת גובה לא שלילי ---
int get_height ()
{
    float height ;
    do {
        printf ("Enter Height ---> ") ;
        scanf ("%f", &height) ;
    } while (height < 0) ;
    return height ;
}

```

שאלה 7:

n	p	s	i	j	פלט	אלגוריתם א'
2	3	0			18	א. טבלת מעקב. ב. ההשמה התבצעה 6 פעמים.
		3	1	1		
		6		2		
		9		3		
		12	2	1		
		15		2		
		18		3		

n	p	s	k	פלט	אלגוריתם ב'
2	3	0		18	א. טבלת מעקב. ב. ההשמה התבצעה 3 פעמים.
		3	1		
		6	2		
		9	3		

האלגוריתם מחשב את $n * p^2$

ג. אלגוריתם ב יותר יעיל כי הוא מבצע פחות איטרציות ("סיבובי לולאה").

שאלה 8:

```

//--- הצפנת מחרוזת: ביטול תווי הרווח, ---
//--- אחרי כל תו הגדול מ-M שותלים את התו הקודם לה ---
//--- אחרי כל תו הקטן מ-M שותלים את התו העוקב לה ---

#include <stdio.h>
#include <string.h>

#define N 120

void main ()
{
    char st [N], new_st [2*N] = {""} ;
    int i, j, k ;

    puts ("Enter a string → ") ;
    gets (st) ;
    k = strlen (st) ;
    j = 0 ; // index for new string
    for (i=0 ; i<k ; i++)
        if (st[i] != ' ') // if not a blank char
        {
            new_st [j++] = st[i] ; // first add 1st char
            if (st[i] < 'M')
                new_st [j++] = st[i] + 1 ; // add successor char
            else
                new_st [j++] = st[i] - 1 ; // add predecessor char
        }
    new_st [j] = '\0' ; // close string
    puts (new_st) ; // printing new string
}

```

פרק ג

שאלה 9:

מערך דו-ממדי המכיל מספרים שלמים: m (18*25)

- קלט למטריצה
- מציאת איבר המינימום במטריצה ← לעמודה min_col, לשורה min_line
- הדפסת איברי המטריצה משורה min_line ועד שורה אחרונה ומעמודה min_col ועד עמודה אחרונה (מאיבר המינימום). עבור כל איבר יש לבדוק אם הוא זוגי. אם כל האיברים זוגיים יודפס "זוגי".

```

//--- קלט מספרים למטריצה, מציאת המינימום ---
//--- והדפסת הריבוע הנוצר מימין ומתחת לאיבר המינימום ---

#include <stdio.h>

#define Line 6 // should be 18
#define Col 8 // should be 25

#define TRUE 1
#define FALSE 0

typedef int mat_type [Line][Col];

void mat_kelet ( mat_type m );
void mat_pelet ( mat_type m, int min_line, int min_col );
void main ()
{
    mat_type m ;
    int i, j, min ;
    int min_line, min_col ;
    mat_kelet ( m );
    //--- looking for smallest element in matrix ---
    min = m [0][0] ;
    min_line = min_col = 0 ;
    for (i=0 ; i<Line ; i++)
        for (j=0 ; j<Col ; j++)
            if (m[i][j] < min)
            {
                min = m[i][j] ;
                min_line = i ;
                min_col = j ;
            }
    mat_pelet ( m, min_line, min_col );
}
    
```

ניתן להפוך קטע זה לפונקציה המקבלת כפרמטר את המערך, ומצביע למשתני השורה ועמודה (ב-C++ עם &). הפונקציה תחזיר במשתנים את מספר התא (שורה ועמודה) של הערך המינימלי.

```
//--- filling matrix with number ---  
//--- במבחן לא היה צריך לקלוט המספרים למטריצה ---  
void mat_kelet ( mat_type m)  
{  
    int i, j;  
    for (i=0 ; i<Line ; i++)  
        for (j=0 ; j<Col ; j++)  
            {  
                printf {"Enter a number → "};  
                scanf ("%d", &m[i][j]);  
            }  
}  
  
//--- הפונקציה מקבלת מטריצה, מספר שורה ומספר עמודה ---  
//--- ומדפיסה איברי המטריצה מנקודה זו ועד פנה ימנית תחתונה של המטריצה ---  
void mat_pelet ( mat_type m, int min_line, int min_col)  
{  
    int zugl = TRUE ;  
    int i, j;  
    for (i = min_line ; i < Line ; i++)  
    {  
        for (j = min_col ; j < Col ; j++)  
        {  
            printf ("%5d", m[i][j]);  
            if (m[i][j]%2 != 0)  
                zugl = FALSE ;  
        }  
    }  
    if (zugl)  
        printf ("All small matrix are pair \n");  
}
```

שאלה 10:

טיפוס תלמיד (stud_type) = מבנה / רשומה

	name	שם = מחרוזת.
	addr	כתובת = מחרוזת.
	id	מס' זיהוי = מספר ארוך.
{ שפת C = 1 , שפת פסקל = 2 }	prog_leng	שפת תכנות = מספר
	grad	ציון = מספר שלם.

stud = מערך בגודל 1750 שכל אחד מאבריו מטיפוס תלמיד.

- קלט נתוני התלמידים למערך stud
- חישוב הציון הממוצע ← avg
- פיצול התלמידים שקיבלו ציון גבוה מהממוצע לשני מערכים ← pas , C
- הדפסת מערך C (שבו j תלמידים)
- הדפסת מערך pas (שבו k תלמידים).

מערכים C ו-pas אף הם בגודל 1750 כל אחד.

```
//--- תחרויות ארציות במדעי המחשב: ---
//--- קלט: פרטי התלמידים ---
//--- פלט: רשימות הגבוהים מהממוצע לפי שפת תכנות ---
```

```
#include <stdio.h>
#define N 6 // should be 1750
typedef struct
{
    char name [30] ;
    char addr [50] ;
    long id ;
    int prog_leng ;
    int grad ;
} stud_type ;
typedef stud_type arr_type [N] ;
void arr_kelet (arr_type stud) ;
void arr_select (arr_type stud) ;
void arr_pelet (arr_type arr, int n) ;
float arr_avg_grad (arr_type stud) ;
```

```
void main ()
{
    arr_type stud ;
    arr_kelet (stud) ;
    arr_select (stud) ;
}

//--- קלט פרטי התלמידים למערך ---
void arr_kelet (arr_type stud)
{
    stud_type T ;
    int i ;
    for (i=0 ; i<N ; i++)
    {
        printf ("Name ---> ") ;
        scanf ("%s",T.name) ;
        printf ("Address ---> ") ;
        scanf ("%s",T.addr) ;
        printf ("ID num ---> ") ;
        scanf ("%ld", &T.id) ;
        printf ("Programming language (1 for C, 2 for Pascal) ---> ") ;
        scanf ("%d", &T.prog_leng) ;
        printf ("grade ---> ") ;
        scanf ("%d", &T.grad) ;

        stud [i] = T ;
        printf ("\n") ;
    }
}

//----- מיון התלמידים שהצליחו למערך פסקל ו-C -----
//--- הפונקציה מקבלת מערך מאותחל של פרטי התלמידים ---
//--- ממיינת את אילו שציוניהם מעל הממוצע למערך פסקל ומערך C ---
//--- ומדפיסה אותם לפי שפה. ---

void arr_select (arr_type stud)
{
    arr_type pas, C ;
    int i, j=0, k=0 ;
    float avg ;

    avg = arr_avg_grad (stud) ;
```

```

for (i=0 ; i<N ; i++)
{
    if (stud[i].grad > avg)
    {
        if (stud[i].prog_leng == '1')
            C [j++] = stud [i] ;
        else
            pas [k++] = stud [i] ;
    }
}

arr_pelet (C, j) ;
arr_pelet (pas, k) ;
}

//--- הדפסת רשימת התלמידים במערך ---
//--- קלט לפוקציה : מערך וגודלו ---
//--- פלט : שם, כתובת ומספר זיהוי ---

void arr_pelet (arr_type arr, int n)
{
    int i ;
    for (i=0 ; i<n ; i++)
    {
        printf ("name:  %s \n", arr[i].name) ;
        printf ("address: %s \n", arr[i].addr) ;
        printf ("ID: %ld \n", arr[i].id) ;
    }
}

//--- טענת כניסה : מערך פרטי התלמידים ---
//--- טענת יציאה : מוחזר ממוצע הציונים ---

float arr_avg_grad (arr_type stud)
{
    int i ;
    int sum = 0 ;
    for (i=0 ; i<N ; i++)
        sum = sum + stud[i].grad ;
    return (float) sum / N ;
}

```