

מדעי המחשב - 2 יחידות לימוד
פתרון בחינת הבטחות

פרק א

שאלה 1:

a	mul	k	$k \leq a$	b	b זוגי?	פלט
3	1					40
	5	1	T	5	F	
	20	2	T	8	T	
	40	3	T	4	T	
		4	F			

קלט: 3, 5, 8, 4 \Rightarrow

שאלה 2:

	m	x	$x \geq 0$	$50 \leq x \leq 70$	פלט
א.	0				2
		10	T	F	
	1	64	T	T	
	2	51	T	T	
		30	T	F	
		-4	F		

ב. -1 (כל מספר ראשון שלילי בקלט)

ג. -1, 80, 10 \Rightarrow (כל המספרים בקלט קטנים מ-50 או גדולים מ-70, לפחות מספר אחד לא שלילי בהתחלה.)

שאלה 3:

קטע תכנית : בכל תא במערך יושם מספר אקראי בין 5 ל- 12 (כולל).

```
randomize();
count = 0;
for (i = 0 ; i < 36 ; i++)
{
    numArr [i] = random (8) + 5;
    if (numArr[i] == 8)
        count ++ ;
}
printf ( " \n הוגרלו %d מספרים השווים ל-8", count);
```

שאלה 4:

int test (int x, int y , float z);

i. x = test (3, 7, 8.2);	תקין.
ii. t = test (b, b, 2.5);	תקין.
iii. c = test (7.8);	לא תקין. חסרים שני פרמטרים.
iv. m = test (4, 6.25, 9);	לא תקין. סדר הפרמטרים אינו תואם את כותרת הפעולה.
v. if (test (2, 4, 6.1) == 1) { k = 7; }	תקין.

שאלה 5:

דרך א : עבור כל תו במחרוזת שהוא רווח, אם התו שלפניו הוא 'y' נוסף 1 למונה.
 דרך ב : עבור כל תו 'y' במחרוזת, אם התו שאחריו הוא תו רווח, נוסף 1 למונה.

<pre>gets (str); count = 0; for (i = 1 ; i < strlen(str) ; i++) if (str[i] == ' ' && str[i-1] == 'Y') count ++ ; printf ("%d", count);</pre> <p>דרך א :</p>	<pre>gets (str); count = 0; for (i = 0 ; i < strlen(str)-1 ; i++) if (str[i] == 'Y' && str[i+1] == ' ') count ++ ; printf ("%d", count);</pre>
--	---

פרק ב

שאלה 6:

```
// קלט: טמפרטורה יומית בתקופה מסויימת. סיום הקלט בטמפי גבוהה מ- 100 מעלות.  
// פלט: מספר הימים המאקסימלי שעבר בין שתי מדידות טמפי הזחה לטמפי שנמדדה ביום הראשון.  
// אם לא נמדדה שוב טמפי זהה לטמפי של היום הראשון, יוצג -1
```

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

```
void main ()
```

```
{  
    int count = 0, max = -1;  
    int firstTemp, temp;  
    printf ("→ טמפרטורה של היום הראשון? ");  
    scanf ("%d", &firstTemp);  
    printf ("→ טמפרטורה של היום הבא? ");  
    scanf ("%d", &temp);  
    while (temp <= 100)  
    {  
        if (temp != firstTemp)  
            count ++ ;  
        else  
        {  
            if (count > max)  
                max = count;  
            count = 0;  
        }  
        printf ("→ טמפרטורה של היום הבא? ");  
        scanf ("%d", &temp);  
    }  
    if (max != -1)  
        printf ("\n מספר הימים המאקסימלי בין שתי מדידות: %d", max);  
    else  
        printf ("\n", max );  
}
```

שאלה 7:

a

	1	2	3	4	5	6	7	8	9	
	3	6	-12	-8	-37	-6	2	-9	13	.א

n	ans	i	$i < 8$ $i < n-1$	a[i]	a[i+1]	a[i+2]	I $a[i+1] > a[i]$	II $a[i+1] > a[i+2]$	I and II
9	0	1	T	3	6	-12	T	T	T
	1	3	T	-12	-8	-37	T	T	T
	2	5	T	-37	-6	2	T	F	F
		7	T	2	-9	13	F	F	F
		9	F						

a

	0	1	2	3	4	5	6	7	8	.ב
	3	6	-12	-8	-37	-6	2	18	13	

טענת יציאה (לא נדרש בבחינה): קטע הקוד סופר כמה איברים במערך גדולים משני שכניהם (שכן בכל צד).

שאלה 8:

א. פעולה (תת-תכנית):

```
// טענת כניסה: קוטר של פקק תקני ומספר הפקקים שמייצרת מכונה
// טענת יציאה: מספר הפקקים התקינים שייצרה המכונה
// פקק תקין הוא פקק שההפרש בערך מוחלט בינו לבין
// פקק בקוטר תקני קטן מ- 1 מ"מ.
```

```
int check (int diameter, int num)
{
    int count = 0, plugDiameter;
    for (int i = 0 ; i < num ; i++)
    {
        printf ("→ הקש קוטר פקק");
        scanf ("%d", & plugDiameter);
        if (abs(plugDiameter - diameter) <= 1)
            count ++ ;
    }
    return count;
}
```

abs() מחייב את ההכללה הבאה בתכנית: #include <math.h>
--

ב. קטע התכנית:

```
int i, standardDiameter, num, count;
for (i = 1 ; i <= 50 ; i++)
{
    printf ("%d → קוטר תקני של מכונה", i);
    scanf ("%d", & standardDiameter);
    printf ("→ מספר הפקקים שיוצרו במכונה");
    scanf ("%d", & num);
    count = check (standardDiameter, num);
    printf ("מכונה %d ייצרה %d פקקים תקינים", count, i);
}
```

פרק ג

לאזה 9:

למעשה זוהי שאלה של רשומות/מבנים/עצמים. נציג שני פתרונות: פתרון ללא רשומות ופתרון עם רשומות.

פתרון ללא רשומות:

טבלת משתנים:

שם משתנה	טיפוס משתנה	תפקיד המשתנה
p	מערך בגודל 319 של שלמים	תאים 1 - 318 משמשים כמקומות חנייה. תא 0 ישמש כקופה. תוכן מקום חניה - שעה ההגעה של מכונית - מספר שלם. תא המכיל 0 מסמל מקום חניה פנוי. תוכן תא 0 - מספר שלם.
arrivalTime	מספר שלם	שעת ההגעה לחניון (בתחום 6 - 22)
departTime	מספר שלם	שעת עזיבה של החניון (בתחום 7 - 23)
code	מספר שלם	קוד המכונית: 1 - כניסה לחניון. 2 - עזיבת החניון.
carPlace	מספר שלם	מציין את מספר התא בו חונה הרכב (בתחום 1-318)

תת-קצוות:

void parkingInit (int p[]); (p, cash) אתחול-חניון
פעולה המקבלת את מערך החניון p ואת הקופה cash ומאתחלת את הקופה ואת כל איברי המערך ל-0.

void arrivalCar (int p[], int arrivalTime); (p, arrivalTime) כניסת-רכב
טענת כניסה: מערך החניון ושעת ההגעה של כלי רכב (מספר שלם בין 6 ל-22).
טענת יציאה: הפעולה מוצאת מקום חניה פנוי במערך החניון, ומסמנת אותו כתפוס על ידי רישום שעת ההגעה של הרכב.

void departCar (int p[], int departTime, int carPlace); (p, departTime, carPlace, cash) יציאת-רכב
טענת כניסה: מערך החניון, שעת העזיבה של כלי רכב (מספר שלם בין 7 ל-23), מקום החניה (מספר התא במערך (מספר שלם בין 1 ל-318) וקופת החניון.
טענת יציאה: הפעולה מחשבת את עלות החניה, מעדכנת את הקופה ומסמנת את המקום כפנוי.

void parkingClose (int p[]); (p) סגירת-חניון
טענת כניסה: מערך החניון.
טענת יציאה: הדפסת סך כל הכסף שנגבה במשך היום בעבור חניית המכוניות בחניון.
הנחה: החניון ריק ממכוניות.

```
#include <stdio.h>
#define N 319
void parkingInit (int p[]);
void arrivalCar (int p[], int arrivalTime);
void departCar (int p[], int departTime, int carPlace);
void parkingClose (int p[]);
//--- התכנית הראשית ---
void main()
{
    int p[N];
    int code, arrivalTime, departTime, carPlace;

    parkingInit (p);

    printf ("type 1 for car arrival, 2 for car departure ");
    printf ("type -1 to end day → ");
    scanf ("%d", & code);
    while (code != -1)
    {
        if (code == 1)
        {
            printf ("type time of arriving (6 to 22) → ");
            scanf ("%d", & arrivalTime);
            arrivalCar (p, arrivalTime);
        }
        else // code == 2
        {
            printf ("type departure time (7 to 23) → ");
            scanf ("%d", & departTime);
            printf ("type number of car location → ");
            scanf ("%d", & carPlace);
            departCar (p, departTime, carPlace);
        }

        printf ("type 1 for car arrival, 2 for car departure ");
        printf ("type -1 to end day → ");
        scanf ("%d", & code);
    }

    parkingClose (p);
}
```

ניהול חניון מכוניות:
קלט - קוד ושעת הגעה או עזיבה.
פלט - סך ההכנסות ליום זה

```

// פעולה המאתחלת את כל איברי המערך ל-0.
void parkingInit (int p[])
{
    for (int i = 0 ; i < N ; i++)
        p[i] = 0;
}

// טענת כניסה : מערך החניון ושעת ההגעה של כלי רכב (מספר שלם בין 6 ל- 22).
// טענת יציאה : הפעולה מוצאת מקום חניה פנוי במערך החניון, ומסמנת אותו כתפוס על ידי רישום
// שעת ההגעה של הרכב.
void arrivalCar (int p[], int arrivalTime)
{
    int i = 1;
    while (i < N && p[i] != 0)
        i++;
    if (i < N)
    {
        p[i] = arrivalTime;
        printf ("Car is parking in place %d \n", i);
    }
    else
        printf ("Parking is full");
}

// טענת כניסה : מערך החניון, שעת העזיבה של כלי רכב (מספר שלם בין 7 ל- 23) ומקום החניה
// (מספר התא במערך (מספר שלם בין 1 ל- 318)).
// טענת יציאה : הפעולה מחשבת את עלות החניה, מעדכנת את הקופה ומסמנת את המקום כפנוי.
void departCar (int p[], int departTime, int carPlace)
{
    int toPay = (departTime - p[carPlace]) * 14;
    printf ("payment for this car : %d nis \n", toPay);
    p[0] = p[0] + toPay;
    p[carPlace] = 0;
}

// טענת כניסה : מערך החניון.
// טענת יציאה : הדפסת סך כל הכסף שנגבה במשך היום בעבור חניית המכוניות בחניון.
// הנחה : החניון ריק ממכוניות.
void parkingClose (int p[])
{
    printf ("total parking money is: %d \n", p[0]);
}

```

כתרון עם רלוואנט:

הפתרון מורכב משתי רשומות: ParkingPlace המגדירה מקום חניה בודד ו-Parking המגדירה את מגרש החניה.

טבלת משתנים:

שם משתנה	טיפוס משתנה	תפקיד המשתנה
מקום חניה: arrivalTime avail	עצם שתכונותיו: שעת-הגעה - שלם פנוי? - בוליאני	
park: cash p	עצם שתכונותיו: קופה - שלם מערך - של עצמים	מגרש החניה - מערך שכל אחד מאיבריו הוא מטיפוס מקום-חניה.
arrivalTime	מספר שלם	שעת ההגעה לחניון (בתחום 6 - 22)
departTime	מספר שלם	שעת עזיבה של החניון (בתחום 7 - 23)
code	מספר שלם	קוד המכוננית: 1 - כניסה לחניון. 0 - עזיבת החניון.
carPlace	מספר שלם	מציין את מספר התא בו חונה הרכב (בתחום 1-318)

חלוקה לתת-פעולות:

void parkingInit (Parking * park); אתחול-חניון (park)
פעולה בונה המחזירה מערך חניות מאותחל וקופה מאופסת.

void arrivalCar (Parking * park, int arrivalTime); כניסת-רכב (park, arrivalTime)
טענת כניסה: שעת ההגעה של כלי רכב (מספר שלם בין 6 ל- 22).
טענת יציאה: הפעולה מוצאת מקום חניה פנוי במערך החניון, מסמנת אותו כתפוס ורושמת את שעת ההגעה של הרכב.

void departCar (Parking * park, int departTime, int carPlace); יציאת-רכב (park, departTime, carPlace)
טענת כניסה: שעת העזיבה של כלי רכב (מספר שלם בין 7 ל- 23) ומקום החניה (מספר התא במערך (מספר שלם בין 0 ל- 317 כולל).
טענת יציאה: הפעולה מחשבת את עלות החניה, מעדכנת את הקופה ומסמנת את המקום כפנוי.

void parkingClose (Parking park); סגירת-חניון (park)
טענת יציאה: הדפסת סך כל הכסף שנגבה במשך היום בעבור חניית המכוניות בחניון.
הנחה: החניון ריק ממכוניות.

void parkingShow (Parking park); הצג-חניון (park)
פעולה המציגה את מצב החניון.
הערה: פעולה זו לא נדרשה בבחינה.

```
#include <stdio.h>

#define TRUE 1
#define FALSE 0

#define N 319

typedef struct
{
    int arrivalTime ;
    int avail;
} parkingPlace;

typedef struct
{
    int cash;
    parkingPlace p[N];
} Parking;

void parkingInit (Parking * park);
void parkingShow (Parking park);
void arrivalCar (Parking * park, int arrivalTime);
void departCar (Parking * park, int departTime, int carPlace);
void parkingClose (Parking park);

void main()
{
    Parking park;
    int code, arrivalTime, departTime, carPlace;

    parkingInit (&park);

    printf ("type 1 for car arrival, 2 for car departure ");
    printf ("type -1 to end day → ");
    scanf ("%d", & code);
    while (code != -1)
    {
        if (code == 1)
        {
            printf ("type time of arriving (6 to 22) → ");
            scanf ("%d", & arrivalTime);
            arrivalCar (&park , arrivalTime);
            parkingShow(park);
        }
        else // code == 2
        {
            printf ("type departure time (7 to 23) → ");
```

ניהול חניון מכוניות:
קלט - קוד ושעת הגעה או עזיבה.
פלט - סך ההכנסות ליום זה

```

scanf ("%d", &departTime);
printf("type number of car location → ");
scanf ("%d", & carPlace);
departCar (&park , departTime, carPlace);
parkingShow(park);
}

printf ("type 1 for car arrival, 2 for car departure ");
printf ("type -1 to end day → ");
scanf("%d", & code);

}

parkingClose (park);
}

```

//--- פעולה בונה ---
//--- הפעולה מאפסת את הקופה ויוצרת מקומות חניה מוכנים לקליטת רכבים ---

```
void parkingInit (Parking * park)
```

```

{
    park->cash = 0;
    for (int i = 0 ; i < N ; i++)
    {
        park->p[i].arrivalTime = 0;
        park->p[i].avail = TRUE;
    }
}

```

park - משתנה מטיפוס חניון.
park.cash - קופת החניון.
park.p - מערך מקומות החניה.
park.p[i] - מקום חניה בודד.
park.p[i].arrivalTime - שעת ההגעה של רכב לחניה ה- i
park.p[i].avail - האם מקום חניה i פנוי או תפוס

//--- פעולה המציגה את מצב מגרש החניה לאחר כל כניסה או יציאת רכב ---
//--- הערה: פעולה זו לא נדרשה בבחינה. ---

```
public void parkingShow (Parking park)
```

```

{
    for (int i = 0 ; i < N ; i++)
    {
        if (park.p[i].avail == TRUE)
            printf("place #%d Avail: TRUE \n", i);
        else
        {
            printf ("place #%d Avail: FALSE ", i);
            printf (" Arriving Time: %d \n", park.p[i].arrivalTime);
        }
    }
    printf("\n");
}

```

```
//--- פעולה המטפלת בכניסת רכב לחניון. ---  
//--- הפעולה מוצאת מקום חניה פנוי במערך החניון, מסמנת אותו כתפוס ורושמת את ---  
//--- שעת ההגעה של הרכב. ---
```

```
void arrivalCar (Parking * park, int arrivalTime)  
{  
    int i = 0;  
    while (i < N && park->p[i].avail == FALSE)  
        i++;  
    if (i < N)  
    {  
        park->p[i].arrivalTime = arrivalTime;  
        park->p[i].avail = FALSE;  
        printf ("Car is parking in place %d \n", i);  
    }  
    else  
        printf ("Parking is full \n");  
}
```

```
//--- פעולה המטפלת ביציאת רכב מהחניון. ---  
//--- הפעולה מחשבת את עלות החניה, מעדכנת את הקופה ומסמנת את המקום כפנוי ---
```

```
void departCar (Parking * park, int departTime, int carPlace)  
{  
    int toPay = (departTime - park->p[carPlace]. arrivalTime) * 14;  
    printf ("payment for this car : %d nis \n", toPay);  
    park->cash = park->cash + toPay;  
    park->p[carPlace]. arrivalTime = 0;  
    park->p[carPlace] = TRUE;  
}
```

```
//--- פעולה המציגה את סך הכסף שנאסף החניון בסיום יום העבודה ---
```

```
void parkingClose (Parking park)  
{  
    printf ("total parking money is: %d \n", park.cash);  
}
```

שאלה 10:

```

//          תכנית המטפלת במערך. אם אין במערך לפחות 7 תאים רציפים המכילים 0
//          תבצע התכנית הזזות במערך עד אשר יהיו בסופו לפחות 7 תאים רציפים המכילים 0.
#include <stdio.h>

#define N 105

//---          ט.כניסה : מערך a ---
//---          ט.יציאה : מוחזר 1 אם קיימים לפחות 7 תאים רציפים המכילים 0, ו-0 אחרת ---
int seven (int a[])
{
    int zero = 0;
    for (int i = 0 ; i < N ; i++)
    {
        if (a[i] == 0)
            zero ++;
        else
            if (zero < 7)
                zero = 0;
    }
    if (zero >= 7)
        return 1;
    else
        return 0;
}

//---          ט.כניסה : מערך ומספר שלם (בין 1 ו-4) ---
//---          ט.יציאה : כל האיברים במערך זזים k מקומות שמאלה. ---
//---          k האיברים הראשונים "גולשים". k האיברים האחרונים מתאפסים ---
void shift (int a[], int k)
{
    int i;
    for (i = 0 ; i < (N - k) ; i++)
        a[i] = a[i+k];
    for (i = (N - k) ; i < N ; i++)
        a[i] = 0;
}

//---          הדפסת המערך ---
void arrPelet (int a[])
{
    for (int i = 0 ; i < N ; i++)
        printf ("%d  ", a[i]);
    printf ("\n");
}

```

```
//--- התכנית הראשית ---  
void main()  
{  
    int a [N];  
    int k;  
    /* כאן יבוא קטע הקלט למערך */  
    while (seven(a) == 0)  
    {  
        printf ("type a number between 1 to 4 → ");  
        scanf ("%d", &k);  
        shift (a, k);  
    }  
    arrPelet (a);  
}
```