

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 ⇒

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 ⇒ (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).

דרך א: שימוש במחלקה Random:

בתכנית תכתב (לפני המחלקה) השורה: `import java.util.Random;`

```
Random rnd = new Random();
int count = 0;
for (int i = 0 ; i < numArr.length ; i++) // אפשר גם : i < 36
{
    numArr [i] = rnd.nextInt (8) + 5;
    if (numArr[i] == 8)
        count ++ ;
}
System.out.println(" הוגרלו " + count + " מספרים השווים ל-8 ");
```

דרך ב: שימוש ב- `Math.random()`:

```
int count = 0;
for (int i = 0 ; i < numArr.length ; i++) // אפשר גם : i < 36
{
    numArr [i] = (int) (Math.random() * 8) + 5;
    if (numArr[i] == 8)
        count ++ ;
}
System.out.println(" הוגרלו " + count + " מספרים השווים ל-8 ");
```

## שאלה 4:

`public static int test (int x, int y , double z)`

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

```
String str = input.next();
int count = 0;
for (int i = 1 ; i < str.length() ; i++)
{
    if (str.charAt(i) == ' ' && str.charAt(i-1) == 'Y')
        count ++;
}
System.out.println(count);
```

דרך א:

```
String str = input.next ();
int count = 0;
for (int i = 0 ; i < str.length()-1 ; i++)
{
    if (str.charAt(i) == 'Y' && str.charAt(i+1) == ' ')
        count ++;
}
System.out.println(count);
```

דרך ב:

תו רווח אחד

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 הזהה לטמפי שנמדדה ביום הראשון.  
// אם לא נמדדה שוב טמפי זהה לטמפי של היום הראשון, יוצג -1
```

```
public class Yessodot2007 {  
    public static void main (String [] args) {  
        Scanner input = new Scanner (System.in);  
        int count = 0, max = -1;  
  
        System.out.print ("→ טמפרטורה של היום הראשון ?");  
        int firstTemp = input.nextInt ();  
  
        System.out.print ("→ טמפרטורה של היום הבא ?");  
        int temp = input.nextInt ();  
        while (temp <= 100)  
        {  
            if (temp != firstTemp)  
                count ++ ;  
            else  
            {  
                if (count > max)  
                    max = count;  
                count = 0;  
            }  
            System.out.print ("→ טמפרטורה של היום הבא ?");  
            temp = input.nextInt ();  
        }  
        if (max != -1)  
            System.out.println(max + " : מספר הימים המאקסימלי בין שתי מדידות");  
        else  
            System.out.println( max );  
    }  
}
```

שאלה 7:

a

	0	1	2	3	4	5	6	7	8
	3	6	-12	-8	-37	-6	2	-9	13

.א

n	ans	i	$i < 7$ $i < n-2$	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	0	T	3	6	-12	T	T	T
	1	2	T	-12	-8	-37	T	T	T
	2	4	T	-37	-6	2	T	F	F
		6	T	2	-9	13	F	F	F
		8	F						

a

	0	1	2	3	4	5	6	7	8
	3	<b>6</b>	-12	<b>-8</b>	-37	-6	2	<b>18</b>	13

.ב

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.
```

```
public static int check (int diameter, int num)
{
    int count = 0, plugDiameter;
    for (int i = 0 ; i < num ; i++)
    {
        System.out.print ("→ הקש קוטר פקק");
        plugDiameter = input.nextInt ();
        if (Math.abs(plugDiameter - diameter) <= 1)
            count ++ ;
    }
    return count;
}
```

ב. קטע התכנית:

```
int i, standardDiameter, num, count;
for (i = 1 ; i <= 50 ; i++)
{
    System.out.print (i + " → קוטר תקני של מכונה");
    standardDiameter = input.nextInt ();
    System.out.print(" → מספר הפקקים שיוצרו במכונה");
    num = input.nextInt ();
    count = check (standardDiameter, num);
    System.out.println("מכונה " + i + " ייצרה " + count + " פקקים תקינים");
}
```

הערה: בשאלה מתבצע קלט בשני מקומות - ב- main ובפעולה. אפשר להגדיר את אובייקט הקלט פעמיים - ב- main ובפעולה, ואפשר גם להגדירו פעם אחת, במחלקה, כאובייקט קלט סטטי (שיוכר בכל התכנית):

```
static Scanner input = new Scanner (System.in);
```

פרק ג

## שאלה 9:

למעשה זוהי שאלה של רשומות/מבנים/עצמים. נציג שני פתרונות: פתרון פרוצדוראלי ופתרון מונחה עצמים.

## פתרון כרוז בוראוי:

## 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:

**public static void parkingInit (int [] p)** (p) אתחול-חניון

פעולה המאתחלת את כל איברי המערך ל-0.

הערה: בשפת java הפעולה מיותרת שכן מעצם המגדרת המערך מאותחלים כל תאיו ל-0.

**public static 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.

(p, departTime, carPlace) יציאת-רכב

**public static void departCar (int [] p, int departTime, int carPlace)**

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.

**public static 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.

```
public class T9Parking {
```

```
    public static final int N = 319;
```

```
    // פעולה המאתחלת את כל איברי המערך ל-0.
```

```
    public static void parkingInit (int [] p)
```

```
    {
```

```
        for (int i = 0 ; i < p.length ; i++)
```

```
            p[i] = 0;
```

```
    }
```

```
    // מערך החניון ושעת ההגעה של כלי רכב (מספר שלם בין 6 ל-22).
```

```
    // טענת כניסה: הפעולה מוצאת מקום חניה פנוי במערך החניון, ומסמנת אותו כתפוס על ידי רישום
```

```
    // טענת יציאה: שעת ההגעה של הרכב.
```

```
    public static void arrivalCar (int [] p, int arrivalTime)
```

```
    {
```

```
        int i = 1;
```

```
        while (i < p.length && p[i] != 0)
```

```
            i++;
```

```
        if (i < N)
```

```
        {
```

```
            p[i] = arrivalTime;
```

```
            System.out.println("Car is parking in place " + i);
```

```
        }
```

```
        else
```

```
            System.out.println("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.
```

```
    public static void departCar (int [] p, int departTime, int carPlace)
```

```
    {
```

```
        int toPay = (departTime - p[carPlace]) * 14;
```

```
        System.out.println("payment for this car : " + toPay + " nis");
```

```
        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.
```

```
    public static void parkingClose (int [] p)
```

```
    {
```

```
        System.out.println("total parking money is: " + p[0]);
```

```
    }
```

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

```
//--- התכנית הראשית ---
```

```
public static void main(String[] args) {
    Scanner input = new Scanner (System.in);
    int [] p = new int [N];
    int code, arrivalTime, departTime, carPlace;

    parkingInit (p);

    System.out.println("type 1 for car arrival, 2 for car departure ");
    System.out.print ("type -1 to end day → ");
    code = input.nextInt ();
    while (code != -1)
    {
        if (code == 1)
        {
            System.out.print ("type time of arriving (6 to 22) → ");
            arrivalTime = input.nextInt ();
            arrivalCar (p, arrivalTime);
        }
        else // code == 2
        {
            System.out.print ("type departure time (7 to 23) → ");
            departTime = input.nextInt ();
            System.out.print ("type number of car location → ");
            carPlace = input.nextInt ();
            departCar (p, departTime, carPlace);
        }

        System.out.println("type 1 for car arrival, 2 for car departure ");
        System.out.print ("type -1 to end day → ");
        code = input.nextInt ();
    }

    parkingClose (p);
}
}
```

**כתרון אונחה צ'אים:** הפתרון מורכב משתי מחלקות: 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 המכונית: 1 - כניסה לחניון. 0 - עזיבת החניון.
carPlace	מספר שלם	מציין את מספר התא בו חונה הרכב (בתחום 0-317)

## תת-פעולות המחלקה Parking

**אתחול-חניון (p)**  
**public Parkingt ()**  
 פעולה בונה המחזירה מערך חניות מאותחל וקופה מאופסת.

**כניסת-רכב (arrivalTime)**  
**public void arrivalCar (int 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.

**יציאת-רכב (p, departTime, carPlace)**  
**public void departCar (int departTime, int 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.

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

```
// מחלקה המטפלת בעצם מסוג "מקום חניה"
class ParkingPlace {
    //--- properties ---
    private int arrivalTime;
    private boolean avail;

    //--- פעולה בונה למקום חניה ---
    //--- הפעולה מאתחלת את שעת ההגעה ל-0 ומסמנת מקום החניה כפנוי ---
    public ParkingPlace() {
        this.arrivalTime = 0;
        this.avail = true;
    }

    //--- פעולה קובעת המסמנת את שעת הגעת הרכב למקום החניה ---
    public void setArrivalTime (int arrivalTime){
        this.arrivalTime = arrivalTime;
    }

    //--- פעולה מאחזרת המחזירה את שעת הגעת המכונית לחניה ---
    public int getArrivalTime (){
        return this.arrivalTime;
    }

    //--- פעולה קובעת המסמנת את מקום החניה כתפוס ---
    public void setAvail(boolean avail){
        this.avail = avail;
    }

    //--- פעולה מאחזרת המחזירה את מצב החניה - פנוי או תפוס ---
    public boolean isAvail(){
        return this.avail;
    }

    //--- פעולה המחזירה את פרטי מקום החניה ---
    public String toString(){
        if (this.avail)
            return ("Avail: " + this.avail + "\n");
        else
        {
            String str = "Avail: " + this.avail;
            str = str + "   Arrivain Time: " + this.arrivalTime + "\n";
        }
        return str;
    }
}
```

מחלקה 1: ParkingPlace  
מחלקה המטפלת במקום חניה בודד.

// מחלקה המטפלת במגרש החניה

```

class Parking {
    public static final int N = 318;
    //--- תכונות המחלקה ---
    private int cash;
    ParkingPlace [] p = new ParkingPlace [N];
    //--- פעולה בונה ---
    //--- הפעולה מאפסת את הקופה ויוצרת מקומות חניה מוכנים לקליטת רכבים ---
    public Parking (){
        this.cash = 0;
        for (int i = 0 ; i < p.length ; i++)
            p[i] = new ParkingPlace();
    }
    //--- פעולה המציגה את מצב מגרש החניה לאחר כל כניסה או יציאת רכב ---
    //--- הערה: פעולה זו לא נדרשה בבחינה. ---
    public void parkingShow (){
        for (int i = 0 ; i < p.length ; i++)
            System.out.print("place #" + i + " " + p[i].toString());
        System.out.println("\n");
    }
    //--- פעולה המטפלת בכניסת רכב לחניון. ---
    //--- הפעולה מוצאת מקום חניה פנוי במערך החניון, מסמנת אותו כתפוס ורושמת את ---
    //--- שעת ההגעה של הרכב. ---
    public void arrivalCar (int arrivalTime){
        int i = 0;
        while (i < p.length && ! p[i].isAvail())
            i++;
        if (i < p.length)
        {
            p[i].setArrivalTime(arrivalTime);
            p[i].setAvail(false);
            System.out.println("Car is parking in place " + i + "\n");
        }
        else
            System.out.println("Parking is full");
    }
}

```

Parking :2 מחלקה  
מחלקה המטפלת במגרש החניה.

```
//--- פעולה המטפלת ביציאת רכב מהחניון. ---
//--- הפעולה מחשבת את עלות החניה, מעדכנת את הקופה ומסמנת את המקום כפנוי ---
public void departCar (int departTime, int carPlace){
    int toPay = (departTime - p[carPlace].getArrivalTime()) * 14;
    System.out.println ("payment for this car : " + toPay + " nis \n");
    this.cash = this.cash + toPay;
    p[carPlace] = new ParkingPlace();
}

//--- פעולה המציגה את סך הכסף שנאסף החניון בסיום יום העבודה ---
public void parkingClose (){
    System.out.println("total parking money is: " + this.cash);
}

//--- התכנית / פעולה הראשית ---
public static void main(String [] args) {
    Scanner input = new Scanner(System.in);
    Parking park = new Parking();
    int code, arrivalTime, departTime, carPlace;

    System.out.println("type 1 for car arrival, 2 for car departure ");
    System.out.print ("type -1 to end day --> ");
    code = input.nextInt ();
    while (code != -1)
    {
        if (code == 1)
        {
            System.out.print ("type time of arriving (6 to 22) --> ");
            arrivalTime = input.nextInt ();
            park.arrivalCar (arrivalTime);
            park.parkingShow();
        }
        else // code == 2
        {
            System.out.print ("type departure time (7 to 23) --> ");
            departTime = input.nextInt ();
            System.out.print ("type number of car location --> ");
            carPlace = Input.nextInt ();
            park.departCar (departTime, carPlace);
            park.parkingShow();
        }
    }

    System.out.println("type 1 for car arrival, 2 for car departure ");
    System.out.print ("type -1 to end day --> ");
    code = input.nextInt ();
}
```

```

    }
    park.parkingClose ();
}
}

```

## שאלה 10:

```

// תכנית המטפלת במערך. אם אין במערך לפחות 7 תאים רציפים המכילים 0
// תבצע התכנית הזו במערך עד אשר יהיו בסופו לפחות 7 תאים רציפים המכילים 0.
public class Targil10 {
    //--- ט.כניסה: מערך a ---
    //--- ט.יציאה: מוחזר 1 אם קיימים לפחות 7 תאים רציפים המכילים 0, ו-0 אחרת ---
    public static int seven (int [] a)
    {
        int zero = 0;
        for (int i = 0 ; i < a.length ; 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 ---
    public static void shift (int [] a, int k)
    {
        int i;
        for (i = 0 ; i < (a.length - k) ; i++)
            a[i] = a[i + k];
        for (i = a.length - k ; i < a.length ; i++)
            a[i] = 0;
    }
}

```

```
//--- הדפסת המערך ---
public static void arrPelet (int [] a)
{
    for (int i = 0 ; i < a.length ; i++)
        System.out.print (a[i] + " ");
    System.out.println("\n");
}

public static final int N = 105;

//--- התכנית הראשית ---
public static void main(String[] args) {
    Scanner input = new Scanner (System.in);
    int [] a = new int [N];
    int k;
    /* --- כאן יבוא קטע הקלט למערך --- */
    while (seven(a) == 0)
    {
        System.out.print ("type a number between 1 to 4 → ");
        k = input.nextInt ();
        shift (a, k);
    }
    arrPelet (a);
}
}
```