

מדעי המחשב – 2 יחידות לימוד
פתרון בחינת הפארות

פרק ראשון

שאלה 1

א. $\Rightarrow 5, 23, 7, 100, 46, 231$

m	x	i	$i \leq 5$	a	$a < 10 \parallel a > 99$	פלט
0	5	1	T	23	false	2
1		2	T	7	true	
2		3	T	100	true	
		4	T	46	false	4
3		5	T	231	true	
		6	F			3

ב. עבור הקלט הבא: $\Rightarrow 2, 7, 231$

יהיה הפלט 2 כי יש רק שני מספרים ושניהם לא דו-ספרתיים ולכן יודפס רק ערכו של m

שאלה 2

```

static Random rnd = new Random();

//--- פעולה המחזירה מספר אקראי בין a - 1 ו- b ---
//--- (ידוע ש-a שונה מ-b, לא ידוע מי הגדול מביניהם) ---
public static int rndNum (int a, int b)
{
    int min = Math.min(a, b); // מאיזה ערך להתחיל
    int gap = Math.abs(a - b) + 1; // מספר המספרים האקראיים בתחום

    return rnd.nextInt (gap) + min;
}

```

הערות:

(1) הגרלת מספר באמצעות `Math.random()`

```
return (int) (Math.random() * gap) + min;
```

(2) אם מפעילים את הפעולה הרבה פעמים, יש להגדיר את אובייקט `Random` מחוץ לפעולה.

עלה 3

```

public class Olympic
{
    //--- תכונות המחלקה --
    private String sport;          // שם המשחק
    private String country;       // שם המדינה

    //--- פעולה בונה ---
    public Olympic(String sport, String country)
    {
        this.sport = sport;
        this.country = country;
    }
}

```

עלה 4

```

//--- פעולה המחזירה את הזמן הקצר מבין הזמנים שהתקבלו ---
//--- דרך 1: בדיקת השעות והדקות ---
public static Timer shorter (Timer t1, Timer t2)
{
    //--- בדיקת שעה קטנה יותר ---
    if (t1.getHour() < t2.getHour())
        return t1;
    if (t1.getHour() > t2.getHour())
        return t2;

    //--- השעה שווה, בדיקת הדקות ---
    if (t1.getMin() < t2.getMin())
        return t1;
    return t2;
}

//--- פעולה המחזירה את הזמן הקצר מבין הזמנים שהתקבלו ---
//--- דרך 2: המרת הזמן לדקות ---
public static Timer shorter1 (Timer t1, Timer t2)
{
    int time1 = t1.getHour()*60 + t1.getMin();
    int time2 = t2.getHour()*60 + t2.getMin();

    if (time1 < time2)
        return t1;
    return t2;
}

```

שאלה 5

.א

```

//--- פעולה המחזירה אמת אם כל ספרות המספר התלת-ספרתי ---
//--- שונות זו מזו, ושקר אחרת ---
//--- דרך 1 ---
public static boolean what (int num)
{
    int d1 = num % 10;           // ספרת האחדות
    int d2 = (num / 10) % 10;    // ספרת העשרות
    int d3 = num / 100;          // ספרת המאות

    if (d1 == d2 || d1 == d3 || d2 == d3)
        return false;
    return true;
}

//--- פעולה המחזירה אמת אם כל ספרות המספר התלת-ספרתי ---
//--- שונות זו מזו, ושקר אחרת ---
//--- דרך 2 ---
public static boolean what1 (int num)
{
    int d1 = num % 10;           // ספרת האחדות
    int d2 = (num / 10) % 10;    // ספרת העשרות
    int d3 = num / 100;          // ספרת המאות

    return d1 != d2 && d1 != d3 && d2 != d3;
}

```

.ב

```

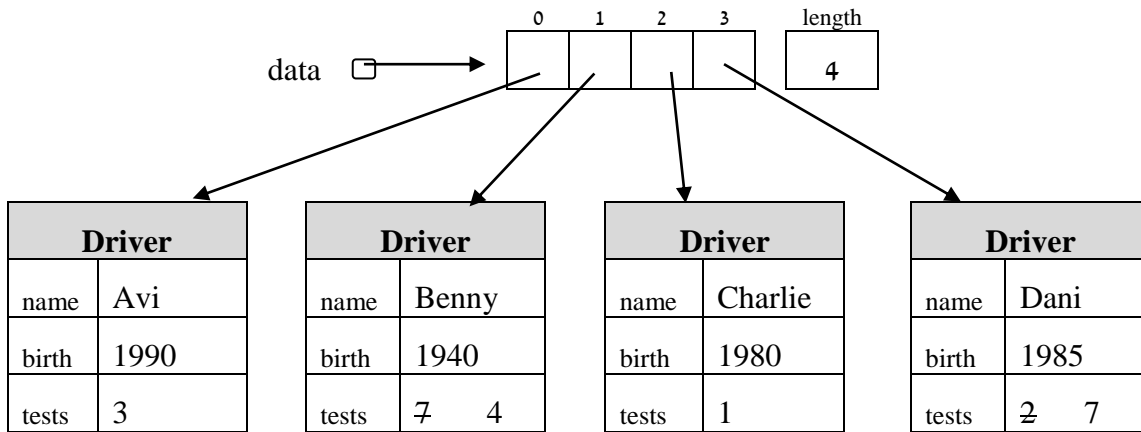
int sum = 0;
for (int i = 0 ; i < data.length ; i++)
{
    if (what (data[i]))
        sum = sum + data[i];
}
System.out.println("sum = " + sum);

```

פרק שני

שאלה 6

.א.



m	x	i	i < 4	d[i].birth	birth > 1984	פלט
0	7	0	T	1990	true	2
1		1	T	1940	false	
		2	T	1980	false	
2		3	T	1985	true	
		4	F			

.ב.

```
//--- פעולה המציגה את שמות כל הנהגים שעברו ---
//--- את מבחן הנהיגה ב - num פעמים או פחות ---
public static void excellentDrivers (Driver [] d, int num)
{
    for (int i = 0 ; i < d.length ; i++)
    {
        if (d[i].getTests() <= num)
            System.out.println(d[i].getName());
    }
}
```

שאלה 7

א.

```

//--- פעול ההמחזירה אמת אם המערך הוא מערך מושלם, ושקר אחרת ---
//--- מערך מושלם הוא מערך שכל אבריו חד ספרתיים חיוביים ---
public static boolean isPerfect (int [] arr)
{
    for (int i = 0 ; i < arr.length ; i++)
        if (arr[i] < 0 || arr[i] > 9)
            return false;
    return true;
}

```

ב. גודל מינימלי של מערך מושלם מלא שיכיל את כל המספרים 1 - 9 (כולל) בדיוק פעם אחת הוא 9.

ג. דרך 1: שימוש בפעולה isExist (arr, digit)

```

//--- פעולה המחזירה אמת אם המערך הוא מערך מושלם מלא ושקר אחרת ---
//--- מערך מושלם מלא מכיל את כל המספרים 1 - 9 (כולל) לפחות פעם אחת ---
public static boolean isPerfectFull (int [] arr)
{
    //--- האם המערך מושלם ? ---
    if (! isPerfect(arr))
        return false;

    //--- האם מכיל את כל הספרות ? ---
    for (int i = 1 ; i <= 9 ; i++)
    {
        if (! isExist(arr, i))
            return false;
    }
    return true;
}

//--- פעולה המקבלת מערך וספרה ומחזירה אמת ---
//--- אם הספרה קיימת במערך, ושקר אחרת ---
public static boolean isExist (int [] arr, int d)
{
    int i = 0;
    while (i < arr.length)
    {
        if (arr[i] == d)
            return true;
        i ++ ;
    }
    return false;
}

```

דרך 2: שימוש במערך עזר:

```
//--- פעולה המחזירה אמת אם המערך הוא מערך מושלם מלא ושקר אחרת ---  
//--- מערך מושלם מלא מכיל את כל המספרים 1 - 9 (כולל) לפחות פעם אחת ---  
public static boolean isFullPerfect (int [] arr)  
{  
    //--- האם המערך מושלם ? ---  
    if (! isPerfect(arr))  
        return false;  
  
    //--- יצירת מערך מונים ---  
    int [] a = new int [10];  
    for (int i = 0 ; i < a.length ; i++)  
        a[i] = 0;  
  
    //--- ספירת מספר המופעים של כל ספרה במערך המקורי ---  
    int d;  
    for (int i = 0 ; i < arr.length ; i++)  
    {  
        d = arr[i];  
        a[d] ++ ;  
    }  
  
    //--- האם כל תאי מערך המונים (פרט לתא 0) גדולים מ-0 ? ---  
    for (int i = 1 ; i < a.length ; i++)  
    {  
        if (a[i] < 1)  
            return false;  
    }  
    return true;  
}
```

שאלה 8

.א.

	0	1	2	3	4	5	6	7	8	9	10	11
arr	14	17	28	36	42	56	72	72	86	100	127	248

x	i	j	$i \leq j$	k	arr[k]	$x \leq arr[k]$	$x \geq arr[k]$	$x == arr[k]$	ערך מוחזר
36	0	12	T	6	72	true	false		
	0	5	T	2	28	false	true		
	3	5	T	4	42	true			
	3	3	T	3	36	true	true		
	4	2	F					true	3

- ב. (1) עבור משפט הזימון `mystery(arr, 100)` יוחזר 9 (כי בתא 9 נמצא הערך 100)
 (2) עבור משפט הזימון `mystery(arr, 50)` יוחזר -1 (כי הערך 50 לא קיים במערך)

ג. הפעולה מבצעת חיפוש בינארי במערך ממויין.

טענת יציאה: אם x קיים במערך יוחזר מקומו, ואם לא קיים יוחזר -1

פרק שלישי

אלף 9

א.

```

//--- הפעולה מקבלת מערך ומספר. אם num נמצא במערך ---
//--- תחזיר הפעולה את מספר האיברים הקטנים ממנו ונמצאים לפניו ---
//--- ואם לא קיים, תחזיר את מספר האיברים הקטנים ממנו במערך ---
public static int foo (int [] a, int num)
{
    int count = 0, i = 0;
    while (i < a.length && a[i] != num)
    {
        if (a[i] < num)
            count ++;
        i ++;
    }
    return count;
}

```

ב. (1) הפעולה החזירה 0:

אם num קיים במערך: כל האיברים לפני גדולים או שווים לו (לא קטנים ממנו).
ואם num לא קיים במערך, כל איברי המערך גדולים ממנו או שווים לא (לא קטנים ממנו).

(2) עבור מערך בגודל 20 החזירה הפעולה 20:

num לא קיים במערך וכל איברי המערך קטנים ממנו.
(אם num היה במקום האחרון במערך, היה מוחזר 19)

ג. עבור מערך ממויין בסדר עולה, מספיק לבדוק את האיברים הקטנים מ-num:

אפשרות 1: שינוי תנאי הלולאה (יעצור מיד אחרי האיבר הראשון)

```
while (i < a.length && a[i] < num)
```

אפשרות 2: לבדוק לפני הלולאה: אם האיבר הראשון גדול מ-num, יוחזר 0

```
if (a[0] < num) return 0;
```

ד. (1) עבור המערך a הנתון תחזיר הפעולה secret את המספר 3.

(2) אם המערך ממויין בסדר עולה, יוחזר גודל המערך.
ואם לא, יוחזר מספרו של התא המפר את המיון.

עאלה 10

.א

```

//--- פעולה המחזירה את סכום ספרותיו של מספר שלם לא שלילי ---
public static int numDigits (int num)
{
    int count = 0;
    while (num > 0)
    {
        num = num / 10;
        count ++;
    }
    return count;
}

```

.ב

```

//--- פעולה המחזירה את מספר המחלקים הזוגיים של המספר ---
public static int pairDivider (int num)
{
    if (num % 2 != 0) // למספר אי-זוגי אין מחלקים זוגיים
        return 0;

    int count = 0;
    int half = num/2;

    for (int i = 1 ; i < half ; i++)
        if (num % i == 0)
            count ++;

    return count;
}

```

פעולה עזר :

```

//--- פעולה המקבלת מספר שלם לא שלילי ---
//--- ומחזירה 1 אם הוא מספר יפה, ו-0 אחרת ---
//--- מספר יפה הוא מספר שסכום ספרותיו ---
//--- שווה למספר המחלקים הזוגיים שלו ---
public static int niceNumber (int num)
{
    if (numDigits(num) == pairDivider (num))
        return 1;
    return 0;
}

```

ג.

```
//--- פעולה המקבלת מערך דו-ממדי ומספר שורה ---  
//--- ומחזירה את מספר המספרים היפים בשורה זו ---  
public static boolean isNiceRow (int [][] mat, int row)  
{  
    int size = mat[row].length; // מספר האיברים בשורה  
  
    int count = 0;  
    for (int i = 0 ; i < size ; i++)  
        count += niceNumber (mat[row][i]);  
  
    return count > size/2; // מספר המספרים היפים גדול ממחצית השורה ?  
}  
  
//--- פעולה המקבלת מערך דו-ממדי ומדפיסה הודעה אם הוא מערך דו-יפה ---  
//--- מערך דו-יפה הוא מערך דו ממדי שבכל שורה יש יותר מספרים יפים ---  
//--- ממספרים שאינם יפים ---  
public static void isNiceMatrix (int [][] mat)  
{  
    boolean nice = true;  
    int i = 0;  
    while (i < mat.length && nice)  
    {  
        if (! isNiceRow(mat, i))  
            nice = false;  
        i ++ ;  
    }  
    if (nice)  
        System.out.println("מערך דו-יפה");  
    else  
        System.out.println("המערך אינו דו-יפה");  
}
```