

פתרון מוצע לבחינת מה"ט ב אלגוריתמיקה ותכנות (java)

מועד א קיץ תש"ף, יולי 2020
מחברת: גב' זהבה לביא, מכללת אורט רחובות

שאלה מספר 1

```
import java.util.Scanner;

public class mainT
{
    /**
     * פתרון שאלה 1
     */
    public static Scanner in = new Scanner (System.in);
    public static void main(String[] args)
    {
        int cnt=0; //סכום המספרים שנקלטו
        int numZugi=0; //סכום המספרים הזוגיים
        int cntPos=0; //סכום המספרים החיוביים
        int totPos=0; //סכום המספרים החיוביים
        int num= in.nextInt();
        while (num!=0) {
            cnt++;
            if (num%2==0)
                numZugi++;
            if (num>0) {
                cntPos++;
                totPos+=num;
            }
            num= in.nextInt();
        }
        System.out.println("סכום מספרים שנקלטו:"+cnt);
        System.out.println("סכום מספרים זוגיים:"+numZugi);
        if (cntPos>0)
            System.out.println("מוצע מספרים חיוביים:"+((double) totPos) / cntPos);
        else
            System.out.println("לא נקלטו מספרים חיוביים");
    }
}
```

שאלה מספר 2

א.

```
public int getInterval() {
    return (23-hour)*60+(60-min);
}
```

ב.

```
public class mainT2
{
    /**
     * פתרון שאלה 2 - סעיף ב
     */
    public static Scanner in = new Scanner (System.in);
    public static void main(String[] args)
    {
        Clock ck= new Clock(in.nextInt(), in.nextInt());
        Clock ck1= new Clock(in.nextInt(), in.nextInt());
        if(ck1.getInterval()>ck.getInterval())
            System.out.println("טיסה ראשונה נוחתת מאוחר יותר");
        else if(ck1.getInterval()<ck.getInterval())
            System.out.println("טיסה שנייה נוחתת מאוחר יותר");
        else
            System.out.println("הטיסות נוחתות באותו הזמן");
    }
}
```

שאלה מספר 3

א.

c	num	ערך מוחזר
0	12345	
1	1234	
1	123	
2	12	
2	1	
3	0	3

1. הפעולה מחזירה את הערך 3.
2. דוגמה למספר שלם וחיובי שעבורו הפעולה מחזירה 0: 2468.
3. הפעולה סופרת את כמות הספרות האי זוגיות לתוך משתנה c ומחזירה את ערכו.

ב.

פלט	<code>if(what(arr[i]<what(res))</code>	<code>what(res)</code>	<code>what(arr[i])</code>	res	i
				1245	
					1
	false	2	3		
					2
	true	2	1	23	
					3
	true	1	0	468	
	false	0	2		4
					5
468					

הפעולה **why** מחזירה את המספר ממערך המספרים, בו כמות הספרות האי זוגיות היא הנמוכה ביותר.

שאלה מספר 4

א.

```

public static boolean isBalanced(int []arr){
    if(arr.length>3 &&arr.length%2==1){
        int midNumm=arr[arr.length/2];
        for(int i=0;i<arr.length/2;i++){
            if(arr[i]<=midNumm)
                return false;
        }
        for(int i=arr.length/2+1;i<arr.length;i++){
            if(arr[i]>=midNumm)
                return false;
        }
    }
    else return false;
    return true;
}
    
```

ב. סיבוכיות הפעולה מסעיף ב: $O(n)$. במקרה הגרוע יתבצע מעבר על כל המערך לבדיקת האיברים מול האיבר האמצעי.

שאלה מספר 5

- א. הפעולה תחזיר **true**.
- ב. הפעולה **one** בודקת האם יש ערך אחד לפחות במערך שחוזר על עצמו. אם כן, מוחזר **true**, אחרת מוחזר **false**.
- ג. הסיבוכיות היא $O(n^2)$. סכום מצטבר של סריקת המערך בעבור כל ערך, החל מהאיבר העוקב לו.
- ד. כן, ניתן לכתוב בצורה יעילה יותר:

```
for(int i=0; i<arr.length-1;i++)
{
    if(arr[i]==arr[i+1])
        return true;
}
return false;
}
```

אין צורך לסרוק את כל המערך בחיפוש ערך זהה. אם קיים כזה, הוא יופיע מיד אחרי התא הנבדק.
 ה. לא, זה לא משנה כיצד המערך ממויין.

שאלה מספר 6

א.

1. דוגמה למחרוזת שתחזיר **true**: **ABCDEFGF**
 2. דוגמה למחרוזת שתחזיר **false**: **EFG3ABC**
 3. הפעולה **why** בודקת אם ישנו תו אחד לפחות במחרוזת המתקבלת שאינו אות אנגלית גדולה. אם תו כזה, מוחזר **false**, אחרת מוחזר **true**.
 4. במקרה הגרוע מתבצעת סריקה של כל המחרוזות ולכן הסיבוכיות הינה: $O(n)$.
- ב.

1. הפעולה רצה במקביל על מערך המחרוזות, איבר מהצד השמאלי של המערך ואיבר מהצד הימני ובודקת אם המחרוזות הנבדקות במקביל שתיהן מכילות רק תווים שהן אותיות אנגליות גדולות או גם תווים אחרים.
- אם הבדיקה מחזירה **true** או **false** בעבור צמד מחרוזות אחד לפחות – יוחזר **false**.

2. דוגמה למערך שיחזיר **true**:

aBc	ZEh	ABC	Ds21	W3EQ	FCV
-----	-----	-----	------	------	-----

הערכים המוחזרים על ידי בדיקת כל אחת מהמחרוזות במקביל תחזיר ערך שונה.

2. דוגמה למערך שיחזיר **false**:

aBc	ZEh	ABC	Ds21	WQEQ	3fC
-----	-----	-----	------	------	-----

הבדיקה של המחרוזות aBc ו- 3fC תחזיר **false** ולכן יוחזר **false** על ידי הפעולה **mystery**.

4. סיבוכיות הפעולה: במקרה הגרוע מתבצע ריצה במקביל על המערך – $n/2$ איברים. הפעולה **why**

ריצה $O(n)$ ומאחר ומתבצעת לולאה (כפל זמן ריצה) – $O(n^2)$

שאלה מספר 7

א.

```
public class Truck
{
    private String truckId;
    private String driverName;
    private int numStorage;
    private boolean isFree;
    public Truck(String truckId, String driverName, int numStorage)
    {
        this.truckId = truckId;
        this.driverName = driverName;
        this.numStorage = numStorage;
        this.isFree = true;
    }
}
```

ב.

```
public static void printNames(Truck [] vec){
    for(int i=0;i<vec.length; i++){
        if(vec[i].isFree() && vec[i].getNumStorage() >=7)
            System.out.println(vec[i].getDriverName());
    }
}
```

ג.

```

public static String getIdFree(Truck [] vec){
    String id="unable to service";
    int maxNumStorage=-1;
    for(int i=0;i<vec.length; i++){
        if(vec[i].isFree()){
            if(vec[i].getNumStorage()>maxNumStorage){
                id=vec[i].getTruckId();
                maxNumStorage = vec[i].getNumStorage();
            }
        }
    }
    return id;
}
    
```

שאלה מספר 8

- א. הפעולה **secret** תחזיר **false**, מאחר והתנאי מתקיים במקרה שבו נבדק הערך האחרון במערך **a** עם הערך השני במערך **b**.
- ב. דוגמה למערכים שיחזירו תוצאה שונה: **a={3,6,9,5}** **b={10,13,15}**
- ג. הפעולה **secret** בודקת האם האיברים במערך **a** קטנים כולם מהאיברים במערך **b**. אם כן, יוחזר **true**, אחרת יוחזר **false**.
- ד. סיבוכיות הפעולה היא : $O(n^2)$. בעבור כל איבר מהמערך הראשון נבדק המערך השני.

שאלה מספר 9:

א.

 מערך **res**:

<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>
<u>3</u>	<u>2</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>3</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>

פלט:

A A A

B B

D

F F F

K

ב. הפעולה סופרת את כמות המופעים של כל אחד מהתווים באמצעות מערך מונים בגודל 26 ומדפיסה כל אות, שמופיעה לפחות פעם אחת, בצורה רציפה לפי מספר המופעים שלה, בשורות נפרדות.

שאלה מספר 10:

א. פלט של המחלקה:

0 1

1 -1

2 1

3 -1

4 1

5 -1

6 1

7 -1

0 5

1 -3

2 8

3 3

4 6

5 7

6 1

7 -1

0 -1
 1 1
 2 7
 3 6
 4 3
 5 8
 6 -3
 7 5

ב. הערה: מאחר ולא ניתן להחזיר יותר מערך אחד על ידי פעולה, תוחזר מחזורית, המכילה את שני הערכים (ערך מינימלי וערך מקסימלי).

```

public String minMavValues(){
    int max, min;
    max=min=this.arr[0];
    for(int i=0;i<arr.length; i++){
        if( arr[i]<min)
            min=arr[i];
        else if (arr[i]>max)
            max = arr[i];
    }
    return min+" ", "+max;
}
    
```

שאלה מספר 11:

א.

1. תוכן המערך לאחר זימון הפעולה: `arr={ 12,678,45,123,345 }`
2. דוגמה למערך בגודל של 5 תאים, כך שהתוכן לא ישתנה לאחר ביצוע הפעולה: 22,78,36,129, 45
3. הפעולה מסדרת את המערך המתקבל כך שבחלקו השמאלי יופיעו המספרים הזוגיים ואח"כ המספרים האי זוגיים (לא לפי הסדר המקורי).

ב.

1. תוכן המערך לאחר זימון הפעולה: `arr={12,34,46,55,67}`
 2. הפעולה מסדרת את המערך כך ש: המספרים הזוגיים יופיעו בתחילה (משמאל לימין) ואח"כ האי זוגיים (תוך הקפדה על הסדר המקורי)
 3. כאשר המערך מסודר מלכתחילה (כל הערכים הזוגיים משמאל ואח"כ הערכים האי זוגיים) זימון שתי הפעולות יביא לאותה התוצאה.
 4. סיבוכיות זמן הריצה של הפעולה **order: O(n)**. סכום סריקות של המערך, כאשר כל סריקה היא **O(n)** ולכן סיבוכיות זמן הריצה **O(n)**.
- סיבוכיות זמן הריצה של הפעולה **arrange: O(n)**. סריקה של המערך פעם אחת.

שאלה מספר 12

א.

```
public class Familiy
{
    private String name;
    private int num;
    private double totalSum;
    public Familiy(int num)
    {
        this.num = num;
        this.totalSum = 0;
    }
    /**
     * @return the totalSum
     */
    public double getTotalSum()
    {
        return this.totalSum;
    }
    public void input() {
        int age;
        double tot=100;
        for(int i=0;i<this.num;i++){
            age=in.nextInt();
            if(age<=3)
                tot+=20.5;
            else if (age >3 && age <=12)
                tot+=30;
            else tot+=40.5;
        }
        this.totalSum = tot;
    }
}
```

ב.

```
public class mainT12
{
    /**
     * פתרון שאלה 12
     */
    public static Scanner in = new Scanner(System.in);
    public static void main(String[] args)
    {
        String name = in.next();
        int num;
        while (!name.equals("STOP")) {
            num=in.nextInt();
            Familiy f = new Familiy(num, name);
            f.input();
            System.out.println("family name:"+name+"
totalSum:"+f.getTotalSum());
            name = in.next();
        }
    }
}
```

שאלה מספר 13

- א. תוצאת הזימון : "ABCDabe"
- ב. דוגמה למחרוזת s2: "A4B23C7"
- ג. דוגמה למחרוזת s3: "A4B2c3Cb7a"
- ד. כן, תיתכן מחרוזת כזו. היא תורכב מאותיות גדולות, קטנות ותווים אחרים, לדוגמה:
"A!B@2c37Cb7*a"
- ה. לא קיימת מחרוזת כזו מאחר וכאשר הפעולה נתקלת בתו שהוא אות גדולה הוא יופיע בתחילת המחרוזת המוחזרת.
- ו. הפעולה יוצרת מחרוזת חדשה מהמחרוזת המתקבלת, המכילה בתחילתה את האותיות הגדולות לפי הסדר שלהן במחרוזת המקורית (אם ישנן כאלה) ואח"כ אותיות קטנות בספר הפוך לסדר שלהן במחרוזת המקורית (אם ישנן כאלה).
- ז. לא כל ערך שלם מתאים לזימון הפעולה, במקרה של ערך שלילי או ערך גדול מאורך המחרוזת המתקבלת נקבל שגיאת זמן ריצה. יש לבדוק שערכו של k אינו שלילי או גדול מאורך המחרוזת בתנאי העצירה (במקרה זה תוחזר מחרוזת ריקה).