

פתרון מוצע לבחינת מה"ט באלגוריתמיקה ותכנות ב-JAVA

מועד א' תשפ"ב, יולי 2022

מחבר: שי אוזן, מכללת אורט רחובות

חלק א'

שאלה מס' 1

```
String str=in.next();
int count=0;
while(str.charAt(0)=='A' &&
      str.charAt(str.length()-1)=='A')
{
    if(str.charAt(0)=='A' ||
       str.charAt(str.length()-1)=='A')
        count++;
    str=in.next();
}
System.out.println("the number of String are:
"+count);
```

שאלה מס' 2

.א

arr	num	left	right	while	if1	if2	temp
18 10 3 12 17 1	4						
		0	5				
				T			
					F	F	
1 10 3 12 17 18							18
				T			
		1			T		
				T			
			4		F	T	
				T			

			3		F	T	
arr	num	left	right	while	if1	if2	temp
				t			
			2		F	T	
				T			
					F	F	
1 3 10 12 17 18							10
				T			
		2			T		
				F			

ב. num = 20

ג. מסדרת את כל המספרים שקטנים מ-num בצד שמאל של המערך ואת כל המספרים שגדולים מ-num בצד ימין של המערך.

שאלה מס' 3

```

8. public static boolean Meuzan(int[] arr)
{
    int countpos=0,countneg=0,countzero=0;
    for (int i = 0; i < arr.length; i++)
    {
        if(arr[i]>0)
            countpos++;
        if(arr[i]<0)
            countneg++;
        if(arr[i]==0)
            countzero++;
    }
    if(countzero == 0 && countpos == countneg)
        return true;
    else
        return false;
}

```

ב.  $O(n)$  כיוון שאני עובר על המערך פעם אחת.

שאלה מס' 4

```

8. public static void printByYear(Child[]arr ,int year)
{
    int boys=0,girls=0;
    for (int i = 0; i < arr.length; i++)
    {
        if(arr[i].getYear()==year)
        {
            if(arr[i].getGender()=='b')
                boys++;
            if(arr[i].getGender()=='g')
                girls++;
        }
    }
    System.out.println("the number of boys: "+ boys);
    System.out.println("the number of girls: "+ girls);
}

2. public static void printByMonth(Child[]arr)
{
    for (int i = 1; i <= 12; i++)
    {
        int boys=0,girls=0;
        for (int j = 0; j < arr.length; j++)
        {
            if(arr[j].getMonth()==i)
            {
                if(arr[j].getGender()=='b')
                    boys++;
                if(arr[j].getGender()=='g')
                    girls++;
            }
        }
        System.out.println("month: "+i);
        System.out.println("the number of boys: "+ boys);
        System.out.println("the number of girls: "+ girls);
    }
}

```

שאלה מס' 5

א. 1. public class Movie

{

private String name;

private double price;

private int num;

2. public Movie() { }

public Movie(string name, double price, int num)

{

this.name = name;

this.price = price;

this.num = num;

}

a	b	פלט
name: "Avatar" price: 30.0 num: 100		
		3000
name: "Avatar" price: 30.0 num: 50		
	name: "Avatar" price: 20.0 num: 50	
		1000

ב.

ג. `public static String getMostPopularMovie(Movie[] t)`  
`{`  
`Movie max=t[0];`  
`for (int i = 1; i < t.length; i++)`  
`{`  
`if(t[i].getNum()>max.getNum())`  
`max=t[i];`  
`}`  
`return max.getName();`  
`}`

שאלה מס' 6

א.

num1	num2	while()	if1	if2	if3	return
1804	1957					
		T				
180	195		T			
		T				
18	19		T			
		T				
1	1		T			
		F				
						F

ב. הפעולה תחזיר אמת רק אם שני המספרים הם לא בעלי אותם כמות הספרות

ובנוסף הספרה השמאלית ביותר של שניהם שווה.

ג. בודק אם לשני המספרים יש כמות ספרות שונה,

דוגמא להפעלה: where(123,456).

שאלה מס' 7

א. `public void voting()`  
`{`  
`int x=in.nextInt();`  
`while(x!=0)`  
`{`  
`if(votes[x-1]!=-1)`  
`votes[x-1]++;`  
`}`

```

x=in.nextInt();

}
}
2. public void update()
{
    int min=-999;
    for (int i = 0; i < votes.length; i++)
    {
        if(votes[i]!=-1 && (min==999||votes[i]<votes[min]))
            min=i;
    }
    votes[min]=-1;
    numOfPart--;
}
3. public boolean isFinish()
{
    if(numOfPart==3)
    {
        for (int i = 0; i < votes.length; i++)
        {
            if(votes[i]!=-1)
                System.out.println(i+1);
        }
        return true;
    }
    return false;
}

```

ב.

public int carrNumber(int numPass)

t		carrNum	pass	while	if	פלט					
carr max	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> = = 20	0	0	0	0	0					
0	0	0	0	0							
			10	T							
carr max	<table><tr><td>10</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> = = 20	10	0	0	0	0	0			F	Your seats in 0
10	0	0	0	0							
			2	T							
carr max	<table><tr><td>12</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> = = 20	12	0	0	0	0	0			F	Your seats in 0
12	0	0	0	0							
			12	T							
carr max	<table><tr><td>12</td><td>12</td><td>0</td><td>0</td><td>0</td></tr></table> = = 20	12	12	0	0	0	1			F	Your seats in 1
12	12	0	0	0							
			4	T							
carr max	<table><tr><td>16</td><td>12</td><td>0</td><td>0</td><td>0</td></tr></table> = = 20	16	12	0	0	0	0			F	Your seats in 0
16	12	0	0	0							
			15	T							
carr max	<table><tr><td>16</td><td>12</td><td>15</td><td>0</td><td>0</td></tr></table> = = 20	16	12	15	0	0	2			F	Your seats in 2
16	12	15	0	0							
			3	T							
carr max	<table><tr><td>19</td><td>12</td><td>15</td><td>0</td><td>0</td></tr></table> = = 20	19	12	15	0	0	0			F	Your seats in 0
19	12	15	0	0							
			15	T							
carr max	<table><tr><td>19</td><td>12</td><td>15</td><td>15</td><td>0</td></tr></table> = = 20	19	12	15	15	0	3			F	Your seats in 3
19	12	15	15	0							
			4	T							
carr max	<table><tr><td>19</td><td>16</td><td>15</td><td>15</td><td>0</td></tr></table> = = 20	19	16	15	15	0	1			F	Your seats in 1
19	16	15	15	0							
			8	T							
carr max	<table><tr><td>19</td><td>16</td><td>15</td><td>15</td><td>8</td></tr></table> = = 20	19	16	15	15	8	4			F	Your seats in 4
19	16	15	15	8							
			2	T							
carr max	<table><tr><td>19</td><td>18</td><td>15</td><td>15</td><td>8</td></tr></table> = = 20	19	18	15	15	8	1			F	Your seats in 1
19	18	15	15	8							
			10	T							
carr	<table><tr><td>19</td><td>18</td><td>15</td><td>15</td><td>18</td></tr></table> =	19	18	15	15	18	4			F	Your seats in 4
19	18	15	15	18							

{

max = 20														
t						carrNum	pass	while	if	פלט				
							3	T						
carr	19	18	18	15	18	=								
max	20					=	2		F	Your seats in 2				
							-99	F						

```

for (int i = 0; i < this.carr.length; i++)
{
    if(this.carr[i]+namPass<maxInCarr)
        return i;
}
return -1;
}

```

שאלה מס' 9

```

8. public static boolean special(String s)
{
    if(s=="")
        return false;
    for (int i = 0; i < s.length(); i++)
    {
        if(!(s.charAt(i)>='a' && s.charAt(i)<='z'))
            return false;
    }
    for (int i = 0; i < s.length()-1; i++)
    {
        for (int j = i+1; j < s.length(); j++)
        {
            if(s.charAt(i)==s.charAt(j))
                return false;
        }
    }
    return true;
}

```

ב.  $O(n^2)$  כיוון שיש לולאה מקוננת שבכל אחת מהן עובר על כל ההמחרוזת.

שאלה מס' 10



א.

1	2	3	4	5
6	7	10	11	12
13	14	15	16	17

- ב. הטווח אמור להיות בין 0 ל 7, כיוון שהוא חייב להיות מ 0 ומעלה וכל תא חייב להיות קטן מהתא שמתחתיו ובנוסף קטן מהאיבר שמימינו.
- ג. הטווח אמור להיות גדול מ 13, כיוון שכל תא חייב להיות גדול מהתא שמעליו ובנוסף גדול מהאיבר שמשמאלו.

### שאלה מס' 11

א. `public static int[] buildK(int[] a, int k)`

```
{
    //יצרנו את המערך המוגדל
    int[] result= new int[a.length*k];
    //עבור בצורת אינדקס על המערך הגדול (החדש) w המשתנה
    int w=0;
    //עובר על המערך הקטן
    for (int i = 0; i < a.length; i++)
    {
        //פעמים על כל אחד מהתאים של k הלולאה הפנימית עוברת
        המערך הקטן
        for (int j = 0; j < k; j++)
        {
            // פה אני מכניס את המספר מהמערך הקטן למערך
            פעמים k הגדול
            result[w]=a[i];
            מתקדם לאיבר הבא במערך הגדול
            w++;
        }
    }
    return result;
}
```

ב. כיוון שהמקסימום הוא שאני עובר על המערך הגדול  $O(n)$  הסיבוכיות היא //

```
public static int isCopyK(int[] arr, int[] brr)
{
```

```

int k = brr.length/arr.length;
int w=0;
for (int i = 0; i < arr.length; i++)
{
    for (int j = 0; j < k; j++)
    {
        if(brr[w]!=arr[i])
            return 0;
        w++;
    }
}
return k;
}

```

שאלה מס' 12

8. 1. public Cake(String name, int day, int month, int year, int price, boolean hasMilk)

```

{
    this.name=name;
    this.lastDate= new Date(day, month, year);
    this.price=price;
    this.hasMilk=hasMilk;
}

```

2. public boolean isFresh(Date d)

```

{
    if(d.before(this.lastDate)==true)
        return true;
    return false;
}

```

3. public boolean isReplaceable(Cake other)

```

{
    int dif=Math.abs(this.price-other.getPrice());
    if(this.lastDate.equals(other.lastDate) &&
        this.hasMilk==other.isHasMilk() && dif<=10)
        return true;
    return false;
}

```

2. 1. public Cake mostFresh()

```

{
    if(numOfCakes==0)
        return null;
    int i=0;
    for (int j = 0; j < numOfCakes; j++)
    {
        if(arr[i].getLastDate().before(arr[j].getLastDate()))
            i=j;
    }
    return arr[i];
}

2. public void removeNotFresh(Date d)
{
    int i=0;
    while(i<numOfCakes)
    {
        if(d.before(arr[i].getLastDate()) ||
           d.equals(arr[i].getLastDate()))
            i++;
        else
        {
            arr[i]=null;
            arr[i]=arr[numOfCakes-1];
            arr[numOfCakes-1]=null;
            numOfCakes--;
        }
    }
}

```

```

3. public int purchaseOrder(String[] names)
{
    int totalPrice=0;
    String detail="";
    String cantBuy="";
    boolean found;
    for (int i = 0; i < names.length; i++)
    {
        found=false;
        for (int j = 0; j < numOfCakes; j++)
        {
            if(arr[j].getName()==names[i])
            {
                found=true;
                detail+=arr[i].toString()+"\n";
                totalPrice+=arr[j].getPrice();
            }
        }

        if(found==false)
            cantBuy+=names[i]+"\\n";
    }

    if(cantBuy=="")
    {
        System.out.println(detail);
        return totalPrice;
    }
    else
    {
        System.out.println(cantBuy);
        return 0;
    }
}

```

א.

s	i	m	j	if	return
VACCINE	6	0			
		1	1	T	
			2	F	
			3	F	
			4	F	
			5	F	
			6	F	
					0

ב.

s	i	r	j	return
VACCINE	3	""		
		C	0	
		CC	1	
		CCA	2	
		CCAV	3	
		CCAVINE		
				CCAVINE

ג.

s	i	m	return
VACCINE	6		
		1	
AVCCINE			
ENICCVA			
	5		
		3	
CINECVA			
VCENICA			
	4		
		1	
CVENICA			
INEVCCA			
	3		
		2	

s	i	m	return
ENIVCCA			
VINECCA			
	2		
		1	
IVNECCA			
NVIECCA			
	1		
		0	
NVIECCA			
VNIECCA			
	0		
		0	
VNIECCA			
VNIECCA			
			VNIECCA

ד. הפעולה מחזירה את המחרוזת מהאות הגדולה לקטנה

### שאלה מס' 14

```

8. public static boolean spot(int[][] mat,int value)
{
    for (int i = 0; i < mat.length-1; i++)
    {
        for (int j = 0; j < mat[i].length-1; j++)
        {
            if(mat[i][j]==mat[i+1][j] &&
               mat[i][j]==mat[i][j+1] &&
               mat[i][j]==mat[i+1][j+1] &&

```

```

        mat[i][j]==value)
            return true;
    }
}
return false;
}

```

ב. `public static boolean dirty(int[][] mat)`

```

{
    int count=0;
    for (int i = 0; i <=9; i++)
    {
        if(spot(mat,i))
            count++;
        if(count>=3)
            return true;
    }
    return false;
}

```

ג. סעיף א  $O(n^2)$  – כיוון שהפעולה עוברת על כל התאים במערך דו מימדי.  
 סעיף ב  $O(n^2)$  - כיוון הפעולה הזו מפעילה את הפעולה מסעיף א מס' קבוע של פעמים.