

## פתרון שאלות דוגמת בחינת בגרות במדעי המחשב א' 2014

### פרק ראשון:

#### שאלה 1:

השלמת פעולה.

פתרונות שגויים אפשריים:

א- מדפיס במקום מחזיר ערך.

לא מבחין בין פעולה שמדפיסה לפעולה שמחזירה ערך

ב- מתייחס להחזרת ערך אמת ולא מטפל בהחזרת ערך שקר.

לא לוקח בחשבון שפעולה בוליאני צריכה להחזיר ערך בכל מקרה או אמת או שקר.

ג- בהשוואה כותב אי שוויון חזק, כלומר גדול ממש ולא גדול או שווה.

בעייה של הבנת הנקרא, המשמעות של "4 קומות ומעלה"

פתרון נכון:

```
public boolean requiredElevator()
{
    return this.numFloors>=4;
}
```

#### שאלה 2:

השלמת פעולה. במחלקה יש טיפוס מסוג מערך של שלמים.

פתרונות אפשריים:

א- חסר איפוס מונה.

ב- אין החזרת המונה אלא הדפסתו.

ג- מבנה לולאה שגוי.

פתרון נכון:

```
public int numOfSuccesses()
{
    int count = 0;
    for (int index=0; index<this.grades.length; index++)
    {
        if (this.grades[index]>85)
            count++;
    }
    return count;
}
```

#### שאלה 3:

מעקב אחר קוד המשתמש בטיפוס, למעשה ההתייחסות הן לתכונות של הטיפוס.

הוראת תנאי מקונן.

פתרונות שגויים אפשריים ????

פשוט מידי, אלא אם כן התלמיד לא מבין מה המשמעות של getX, getY.

הייתי הופכת את השאלה לקטע נוסף בלי תנאי 36קונן ומבקשת לבדוק אם התלמיד יודע ששני קטעי הקוד אינם שקולים.

**שאלה 4:**

דרגת קושי הרבה יותר גבוהה מדרגת הקושי של תרגילים 1-3 השוואה בין 2 מערכים. הנחה (חסר בשאלה) : המערכים באותו הגודל.

תבנית כל הערכים מקיימים תנאי, תלמידים רבים משתמשים ב if-else ומשנים את הערך הבוליאני.

שגיאה אפשרית 1: "התנדנדות" הערך המוחזר. יחזיר אמת אם הציון האחרון לא יותר נמוך ולא אם כל הציונים לא יותר

נמוכים.

```
public boolean isImproved()
{
    for (int index=0; index<this.gradesSimesterA.length; index++)
    {
        if (this.gradesSimesterA[index]<this.gradesSimesterB[index])
            return true;
        else
            return false;
    }
}
```

**שגיאה אפשרית 2:**

אתחול המשתנה הבוליאני לשקר, במידה ומצאנו זוג ציונים שעונים על התנאי ישתנה לאמת. לא מתאים לבדיקה שכל הציונים צריכים להיות לא נמוכים מהראשונים.

```
public boolean isImproved()
{
    boolean flag = false;
    int index = 0;
    while (index<this.gradesSimesterA.length)
    {
        if (this.gradesSimesterA[index]<=this.gradesSimesterB[index])
            flag = true;
    }
    return flag;
}
```

פתרון נכון: אין צורך לעבור עם 2 אינדקסים על המערכים שכן ההנחה (שחסרה) שהם באותו האורך.

כמו-כן, נכון לעצור ברגע והגענו להחלטה.

```
public boolean isImproved()
{
    int index1=0;
    int index2=0;
    while (index1<this.gradesSimesterA.length && index2<this.gradesSimesterB.length)
    {
        if (this.gradesSimesterB[index2]<this.gradesSimesterA[index1])
            return false;
        index1++;
        index2++;
    }
    return true;
}
```

**שאלה 5:**

הוספת פעולה ראשית אשר משתמשת בטיפוס נתון. הפעולה יוצרת עצם מטיפוס המחלקה ומפעילה פעולות עליו.

שגיאות אפשריות

- הדפסת ערכי משתני הקלט ולא brutto, name, ולא את התכונות של העובד.

- זימון שגוי של הפעולה netto

פתרון נכון:

```
import java.util.Scanner;
public class Run
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        String name; // name of employee
        double brutto; // brutto salary
        double netto; // netto salary
        System.out.println("Enter name of Employee");
        name = in.next();
        System.out.println("Enter brutto salary");
        brutto = in.nextDouble();
        // create object of Employee
        Employee employee = new Employee(name, brutto);
        System.out.println("name of Employee is: " + employee.getName());
        System.out.println("brutto salary is: " + employee.getSalary());
        netto = employee.netto();
        System.out.println("netto salary is: "+ employee.netto());
    }
}
```

**פרק שני:****שאלה 6** השלמת גוף פעולה במחלקה המכילה כתכונה מערך של מחרוזותפתרונות שגויים אפשריים:

- 1- גלישה מגבולות המערך בבדיקה.
- 2- שימוש לא נכון בחזרת אמת או שקר.
- 3- השוואת זוגות במקום כל תא במערך להשוות לכל השאר.

פתרונות נכונים אפשריים:

1) עוברים על כל איבר במערך הצבעים ומשווים את הערך לכל אחד מהבאים אחריו במערך, במידה ומוצאים שקיים, מחזירים שקר. במידה ואין כזה שנמצא יותר מפעם אחת, מחזירים אמת.

```
public boolean isAllDifference()
{
    for (int k=0; k<this.colors.length-1; k++)
    {
        for (int p=k+1; p<this.colors.length; p++)
            if (this.colors[k].equals(this.colors[p]))
                return false;
    }
    return true;
}
```

כנ"ל בלי לבצע return באמצע הלולאה

```
public boolean isAllDifference()
{
    Boolean ok=true;
    for (int k=0; k<this.colors.length-1 && ok; k++)
    {
        for (int p=k+1; p<this.colors.length; p++)
            if (this.colors[k].equals(this.colors[p]))
                ok= false;
        // ok = (this.colors[k].equals(this.colors[p])); צורת פתרון אחרת
    }
    return ok;
}
```

2) יוצרים מחרוזת ריקה שתכיל את כל הצבעים אשר שונים זה מזה.. עוברים על מערך הצבעים, עבור כל צבע במערך, במידה וקיים במחרוזת הצבעים, מחזירים שקר, אחרת מוסיפים אותו למחרוזת הצבעים. מחזירים אמת אם לא יצאנו מהפעולה עם שקר.

```
public boolean isAllDifference()
{
    if (this.colors.length==0)
        return true;
    String colors=this.colors[0];
    for (int k=1; k<this.colors.length; k++)
    {
        if (colors.indexOf(this.colors[k])>=0)
            return false;
        colors+=this.colors[k];
    }
    return true;
}
```

**שאלה 7:** מעקב אחר תכנית, השלמת פעולה

t	id	tickets	tickets>0	id==-1	output				
<table border="1"> <tr><td>Train</td></tr> <tr><td>0   0   0   0   0</td></tr> <tr><td>20</td></tr> </table>	Train	0   0   0   0   0	20	?	10	True			
Train									
0   0   0   0   0									
20									
	1			False					
<table border="1"> <tr><td>Train</td></tr> <tr><td>1   0   0   0   0</td></tr> <tr><td>0</td></tr> <tr><td>20</td></tr> </table>	Train	1   0   0   0   0	0	20					Confirm
Train									
1   0   0   0   0									
0									
20									
		2	True						
	1			False					
<table border="1"> <tr><td>Train</td></tr> <tr><td>1   0   0   0   0</td></tr> <tr><td>2</td></tr> <tr><td>20</td></tr> </table>	Train	1   0   0   0   0	2	20					confirm
Train									
1   0   0   0   0									
2									
20									
		12	True						
	2			False					
<table border="1"> <tr><td>Train</td></tr> <tr><td>1   1   0   0   0</td></tr> <tr><td>2   2</td></tr> <tr><td>20</td></tr> </table>	Train	1   1   0   0   0	2   2	20					Confirm
Train									
1   1   0   0   0									
2   2									
20									
		4	True						
	1			False					
<table border="1"> <tr><td>Train</td></tr> <tr><td>1   1   0   0   0</td></tr> <tr><td>6   2</td></tr> <tr><td>20</td></tr> </table>	Train	1   1   0   0   0	6   2	20					Confirm
Train									
1   1   0   0   0									
6   2									
20									
		15	True						
	3			False					
<table border="1"> <tr><td>Train</td></tr> <tr><td>1   1   1   0   0</td></tr> <tr><td>6   2   5</td></tr> <tr><td>20</td></tr> </table>	Train	1   1   1   0   0	6   2   5	20					Confirm
Train									
1   1   1   0   0									
6   2   5									
20									
		3	True						

t	id	tickets	tickets>0	id==1	output																				
	1			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>9</td><td>2</td><td>5</td><td></td><td></td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	0	0	9	2	5			20									Confirm
Train																									
1	1	1	0	0																					
9	2	5																							
20																									
		15	True																						
	4			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>9</td><td>2</td><td>5</td><td>5</td><td></td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	1	0	9	2	5	5		20									Confirm
Train																									
1	1	1	1	0																					
9	2	5	5																						
20																									
		4	True																						
	2			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>9</td><td>6</td><td>5</td><td>5</td><td></td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	1	0	9	6	5	5		20									Confirm
Train																									
1	1	1	1	0																					
9	6	5	5																						
20																									
		8	True																						
	5			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>8</td></tr> <tr><td>9</td><td>6</td><td>5</td><td>5</td><td></td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	1	8	9	6	5	5		20									confirm
Train																									
1	1	1	1	8																					
9	6	5	5																						
20																									
		2	True																						
	2			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>8</td></tr> <tr><td>9</td><td>8</td><td>5</td><td>5</td><td></td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	1	8	9	8	5	5		20									Confirm
Train																									
1	1	1	1	8																					
9	8	5	5																						
20																									
		10	True																						
	4			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>9</td><td>8</td><td>5</td><td>5</td><td>8</td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	1	1	9	8	5	5	8	20									Confirm
Train																									
1	1	1	1	1																					
9	8	5	5	8																					
20																									
		3	True																						
	3			False																					
<table border="1"> <tr><td colspan="5">Train</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>9</td><td>8</td><td>8</td><td>5</td><td>8</td></tr> <tr><td colspan="5">20</td></tr> </table>	Train					1	1	1	1	1	9	8	8	5	8	20									Confirm
Train																									
1	1	1	1	1																					
9	8	8	5	8																					
20																									
		-99	false																						

## מימוש הפעולה carrNumbers

```
public int carNumber(int numTickets)
{
    for (int k=0; k<this.wagons.length; k++)
    {
        if (this.wagons[k]+numTickets<=this.maxInWagon)
            return (k+1);
    }
    return (-1);
}
```

טענת כניסה: הפעולה מקבלת מספר כרטיסים  
טענת יציאה: הפעולה מחזירה את מספר הקרון הראשון  
 שניתן להוסיף לו נוסעים. (-1) אם אין כזה.

קוד המחלקות עם שינוי שמות פעולות

```
/**
 * Question7:
 * @author evi
 */
public class Train
{
    private int[] wagons; // arrays of wagons
    private int maxInWagon; // maximum passengers in a wagon
    /**
     * create Train with n wagons and in each wagon max passengers
     * @param n: int, number of wagons
     * @param max : int, maximum passengers in a wagon
     */
    public Train(int n, int max)
    {
        this.wagons = new int[n];
        for (int k=0; k<n; k++)
            this.wagons[k]=0;
        this.maxInWagon = max;
    }
    /**
     * add
     * @param wagonID: number of wagon, between 1 to number of wagons
     * @param numTickets : number of tickets, positive and can be added
     * means together less than maximum passenger in a wagon
     */
    public void addToWagon(int wagonID, int numTickets)
    {
        this.wagons[wagonID-1] = this.wagons[wagonID-1]+numTickets;
    }
    /**
     * @param numTickets: int
     * @return number of wagon that can contains these number of passengers
     * (-1) if there in no such one
     */
}
```

```

public int wagonNumber(int numTickets)
{
    for (int k=0; k<this.wagons.length; k++)
    {
        if (this.wagons[k]+numTickets<=this.maxInWagon)
            return (k+1);
    }
    return (-1);
}

@Override
public String toString()
{
    String str="";
    str+= "maximum passengers in a wagon is: "+this.maxInWagon+"\n";
    for (int k=0;k<this.wagons.length; k++)
        str+= "wagon "+(k+1)+": "+this.wagons[k]+" passengers\n";
    return str;
}
}

import java.util.Scanner;
/**
 * Question7:
 * @author evi
 */
public class RunTrain
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        Train t = new Train(5, 20);
        int id;
        System.out.println("Enter number of tickets");
        int tickets = in.nextInt();
        while (tickets>0)
        {
            id = t.wagonNumber(tickets);
            if (id==-1)
                System.out.println("not confirm");
            else
            {
                t.addToWagon(id, tickets);
                System.out.println("confirm");
            }
            System.out.println("Enter number of tickets");
            tickets = in.nextInt();
        }
        System.out.println(t);
    }
}

```



דוגמת פלט:

Enter number of tickets

10

confirm

Enter number of tickets

2

confirm

Enter number of tickets

12

confirm

Enter number of tickets

4

confirm

Enter number of tickets

15

confirm

Enter number of tickets

3

confirm

Enter number of tickets

15

confirm

Enter number of tickets

4

confirm

Enter number of tickets

8

confirm

Enter number of tickets

2

confirm

Enter number of tickets

10

confirm

Enter number of tickets

3

confirm

Enter number of tickets

-99

maximum passengers in a wagon is: 20

wagon 1:19 passengers

wagon 2:18 passengers

wagon 3:18 passengers

wagon 4:15 passengers

wagon 5:18 passengers

**שאלה 8:**

בניית מחלקה שהיא טיפוס מורכב. הטיפוס שיש לבנות מכיל בתוכו מערך של הטיפוס Worker

**הערות:**

- תחושה של שאלה שאולצה להשתנות לשיטה החדשה..
- לא בדיוק ברור מה הכוונה בשאלה. האם מחלקת MonthlySalary קולטת את הנתונים מהמשתמש? האם היא מקבלת מערך עם כל הנתונים? או שהיא מקבלת פרטים רלוונטים עבור כל עובד ומוסיפה למערך?
- כ"כ לא ברור מה לעשות אם יש 0 עובדים או שיש יותר מ-5 עובדים??
- במחלקות לא נכניס הדפסות אלא נחזיר מחרוזת עם הערך המבוקש (כמו ToString) ופה התבקשנו במפורש לנהוג אחרת!
- בשאלה רשום להדפיס את שמות העובדים שמרוויים מעל הממוצע אבל בדוגמא יש את כל פרטי העובדים!

**דילמות:**

האם לכלול בתכונות המחלקה MonthlySalary בנוסף למערך העובדים גם מערך של שכר העובדים, כדי לייעל את העבודה) והשכר הממוצע?  
 מצד אחד זה יפשט את העבודה עם המחלקה, מצד שני, בפעולה הבונה יש לאתחל את כל התכונות. כלומר: להציב שכר 0 ושכר ממוצע 0 לכל העובדים. בעייתי, כשנתייחס לשכר העובדים, יהיה צורך להפעיל קודם את פעולת השכר ושהשכר הממוצע. בקיצור, שגוי.  
 מצד שני, יש צורך לחשב את שכר העובדים לפני חישוב הממוצע ולהשתמש בפעולה זו להדפסת שמות העובדים אשר שכרם מעל הממוצע. חזרה על הפעלת פעולה יותר מפעם אחת, בעוד שאם הנתונים של השכר היו נשמרים כתכונות לא היה צריך לחשב שוב.  
 אפשרות נוספת, להוסיף לעובד תכונה ששומרת את שכר העובד.  
 האם מותר להוסיף תכונה למחלקה שנתנו לנו? סביר להניח שלא.

**פתרון:**

שם המחלקה	MonthSalary
מספר העובדים במפעל מערך עובדים	public final int MAX_WORKERS=5; private Worker[] workers
פעולה יוצאת מפעל עם 5 נתוני עובדים פעולה המחזירה את שכר העובדים פעולה המחזירה את השכר הממוצע פעולה המדפיסה את שמות העובדים עם שכר מעל הממוצע	public MonthlySalary(Worker[] workers) public double[] salary() public double avgSalary() public void printAboveAverage()

טענת יציאה: מחזירה את שכר העובדים במפעל

```
public double[] salary()
{
    double[] salaries = new double[MAX_WORKERS];
    for (int k=0; k<MAX_WORKERS; k++)
    {
        salaries[k] = this.workers[k].salary();
    }
    return salaries;
}
```

```

public double avgSalary()
{// טענת יציאה: הפעולה מחזירה את השכר הממוצע
    double[] salaries = this.salary();
    double avg = 0;
    for (int k=0; k<salaries.length; k++)
        avg += salaries[k];
    avg = avg/MAX_WORKERS;
    return avg;
}

public void printAboveAverage()
{// טענת יציאה: הפעולה מדפיסה את שמות העובדים בעלי שכר מעל ממוצע
    double avg = this.avgSalary();
    System.out.println("the average salary is: "+ avg);
    System.out.println("Names of workers with salary above average");
    for (int k=0; k<MAX_WORKERS; k++)
    {
        if (this.workers[k].salary(>avg)
            System.out.println(this.workers[k].getName());
    }
}
}

```

דוגמה לפלט:

name	hours	payHour	salary
sarit	72	26.5	1908.0
liad	68	33.0	2244.0
doron	60	46.0	2760.0
dana	90	34.5	3105.0
shear	66	38.0	2508.0

The salaries of all workers:

1908.0 , 2244.0 , 2760.0 , 3105.0 , 2508.0 ,

the average salary is: 2505.0

Names of workers with salary above average

doron

dana

shear

## פתרון של דורית:

```

public class MonthlySalary
{
    private Worker[] ary;
    private int count;
    public MonthlySalary()
    {
        ary= new Worker[5];
        count = 0;
    }

    public void addWorker(string n, int h, double pPH)
    {
        if (count<ary.Length )
        {
            Worker w = new Worker ( n, h, pPH);
            ary[count]=w;
        }
        // else          במקרה הזה אין הנחיות למקרה הזה
        // Console.WriteLine("עובדים מדי יותר");
    }
    // פעולה המחשבת את שכר כל העובדים
    public double AllSaleries()
    {
        double sum = 0;
        for (int i = 0; i < ary.Length; i++)
        {
            sum = sum + ary[i].Salary();
        }
        return sum;
    }

    private double avgSalary()
    {
        if (this.count == 0)
            return 0;
        else
            return this.AllSaleries() / this.count;
    }
    public void PrintAboveAvg()
    {
        double avg= avgSalary ();
        for (int i = 0; i < ary.Length; i++)
        {
            if (ary[i].Salary ()>avg)
                Console.WriteLine(ary[i].GetName ()); // ביקשו להדפיס רק את שם העובד
                // אם כי בדוגמא יש את כל פרטי העובדים.
                // לשם כך עדיף להשתמש בפעולה ToString של מחלקת Worker
        }
    }
}

```

**פרק שלישי:****שאלה 9: Competition****הערות של EVI**

- 1) האם אנחנו מאפשרים קלט בתוך הפעולה?
- 2) אין איפוס מערך המונים בפעולה הבונה. – הפעולה הבונה לא כתובה אז...ניתן להניח שהמערך מאותחל !!
- 3) שאלה מנוסחת לא טוב. קשה להבנה ולוקח המון זמן לנתח את הבעיה.  
מצד אחד יש 20 מועמדים אבל צוברים את הקולות לכל תלמידי ב"ס (שם המשתנה מוגדר בצורה מבלבלת) אם שומרים מערך של התלמידים זה רק כדי לוודא שתלמיד לא הצביע יותר מפעם אחת.
- 4) מה קורה עם תקינות קלט? האם יש לטפל?
- 5) הפעולה addVotter צריכה לקבל 3 פרמטרים שהם מספרי המועמדים ולהגדיל את מספר המצביעים והקלט צריך להיות בפעולה הראשית.
- 6) תיעוד הפעולה addVotter, אמנם אין טענת כניסה אבל הפעולה קולטת. צריך לדעתי לציין זאת.  
לדעתי, בכלל לא צריך להיות כך, היא צריכה לקבל 3 פרמטרים שהם מספרי התלמידים הנבחרים.
- 7) הייתי מוסיפה דרישה שחובה לממש את הפעולה addVotter בעזרת הפעולה addPoints.

...

**הערות של דורית:**

- א. השאלה לטעמי מאולצת מבחינת ההסבה לתוכנית החדשה יותר נכון היה, מבחינת כתיבת המחלקה, לקבל 3 מספרים של מתמודדים ולתת להם (במחלקה) את הניקוד - ואז אין צורך בפעולה הנוספת addVotter.
- ב. הבניה עם מספר התלמידים בכיתה - ממש לא שייך. הגיוני יותר שהמחלקה תחזיר כמה הצביעו ובתוכנית נחשב את האחוזים או...שהפעולה GetPercente תקבל את סה"כ התלמידים ותחזיר מה אחוז ההצבעה.
- ג. לא ברור מה קורה אם עבור המועמד השני או השלישי קיבלנו 0.
- ד. עוצרים ו...מה קורה עם ההצבעות שכבר קיבלנו ??
- ה. למה האילוץ בתוך המחלקה שתלמיד במקום 0 ייקבל את מספר 1 ??
- ו. יותר נכון לומר שמספר המועמדים מתחיל מ-1 ולתת לתלמידים שיפתרו כמו שהם רוצים.

**קשיים אפשריים:**

- 1- הבנת הנקרא
- 2- איפה קולטים?
- 3- איך קובעים את הפסקת הקליטה ומה עושים עם זה. 0 – מפסיק אבל איך מטפלים בזה?
- 4- חישוב אחוזים. ישנם תלמידים שמתקשים בטפול באחוזים, זה שייך למתמטיקה ולא למדעי המחשב. לא בודק הבנה של מדעי המחשב.
- 5- איך מטפלים בתקינות הלקט. הפעולה addPoints מניחה קלט תקין אז יש להעביר אליה קלט תקין.
- 6- מתי ואיפה מגדילים את מונה המצביעים?

**תשובות שגויות:****א- לגבי הפעולה הראשית:**

- 1- התלמיד לא בנה את העצם מטיפוס Competition בצורה נכונה. לא הבין מה לעשות עם הקלט של מספר התלמידים.
- 2- התלמיד קלט שלשות של מספרים עבור הצבעות התלמידים אבל לא כתב נכון את הוראת התנאי של הלולאה.
- 3- התלמיד לא טיפל נכון בתנאי הפסקת הלולאה.
- 4- התלמיד לא ידע לזמן נכון את הפעולות למציאת אחוז המצביעים והתלמיד המנצח.  
סביר להניח שאם טעה באחד מהם יטעה בשני כי זו אותו סוג של שגיאה.
- 5- התלמיד לא דאג לבדיקת קלט תקין וכך גרם לחריגה מגבולות מערך.

ב- לגבי הפעולה **addVotter**

- 1- אין בדיקה שהפרמטרים נכונים ואז עלול להיות מצב של גישה לתא במערך שלא קיים. גלישה מגבולות המערך.
- 2- התלמיד שכח להגדיל את מונה המצביעים
- 3- התלמיד הגדיל את מונה המצביעים גם אם אי אפשר להוסיף את נתוני הצבעה שכן הקלט שגוי.
- 4- התלמיד ניגש למיקום לא נכון במערך המצביעים (לא הוריד / הוסיף 1).

בעיות שיכולות לצוץ:

- לא שמו את הזימון לפעולה AddVotter במקום הנכון, אלא אחרי כל הצבעה (ולא אחרי שלישייה)
  - לא קלטו בפעם הראשונה את מס התלמידים + הצבעה ראשונה
  - לא קלטו בסוף הלולאה שוב
  - לא עצרו כשהקלט = 0
  - לא עצרו כשהקלט לבחירה השניה או השלישית הוא 0
  - התייחסו לתלמיד הראשון כמספר 0 (ולא 1) או... שהוסיפו 1 בהכנסה ולא הורידו 1 בהוצאה
- הכל... בעיות שהיו ע"פ התוכנית הישנה בלבד.

פתרון בג'אווה עם שינויי השאלה לטעמי - EVI

```
import java.util.Scanner;
/**
 * @version 17/2/14
 * @author evi
 */
public class Competition
{
    public static final int MAX_CANDIDATES = 20;
    private int[] candidates; // array of candidates
    private int numStudents; // number of students in all school
    private int numVotter; // number of voters
    /**
     * create Competition with number of students
     * @param numStudents : int, number of students
     */
    public Competition(int numStudents)
    {
        this.numStudents = numStudents;
        this.numVotter = 0;
        this.candidates = new int[MAX_CANDIDATES];
        for (int stud=0; stud<this.candidates.length; stud++)
        { // איפוס מערך מונים
            this.candidates[stud]=0;
        }
    }
}
```

```

/**
 * add votes
 * @param stud1:int, first candidate
 * @param stud2:int, second candidate
 * @param stud3 int, third candidate
 * assumption n1,n2,n3 correct number of student
 */
public void addVoter(int stud1,int stud2,int stud3)
{
    if (stud1>=1 && stud1<=MAX_CANDIDATES &&
        stud1>=1 && stud2<=MAX_CANDIDATES &&
        stud1>=1 && stud3<=MAX_CANDIDATES )
    {
        this.addPoints(stud1, 3);
        this.addPoints(stud2, 2);
        this.addPoints(stud3, 1);
        this.numVotter++;
    }
}
/**
 * add points to id
 * @param id: number of student
 * @param points : int
 * assumption: id is correct, points is 1,2 or 3
 */
public void addPoints(int id, int points)
{
    this.candidates[id-1]+= points;
}
/**
 * @return percent of students who voted
 */
public double getPercent()
{
    return (double)this.numVotter/this.numStudents*100;
}
/**
 * @return the serial number of the winner
 * assumption there is only one
 */

```

```

public int getWinner()
{
    int winIndex = 0;
    for (int i=1; i<this.candidates.length; i++)
    {
        if (this.candidates[i]>this.candidates[winIndex])
            winIndex = i;
    }
    return winIndex+1;
}
}

import java.util.Scanner;
/**
 * @version 17/2/14
 * @author evi
 */
* input: number of students at school
* the votes: for each student 3 serial number of candidate
* end with 0
* output: the percent of voters
* the winner
*/
public class RunCompetition
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int numStudents;
        int id1, id2, id3;
        // get number of students
        System.out.println("Enter number of students");
        numStudents = in.nextInt();
        // create Competition with number of students
        Competition c = new Competition(numStudents);
        // get votes
        System.out.println("Enter vote, 3 student's number (1-"+Competition.MAX_CANDIDATES+"");
        id1= in.nextInt();
        while (id1!=0)
        {

```



```

    id2 = in.nextInt();
    id3 = in.nextInt();
    c.addVoter(id1, id2, id3);
    System.out.println("Enter vote, 3 student's number (1-"+Competition.MAX_CANDIDATES+"");
    id1= in.nextInt();
}
// caculates percent of voters
double percentVotters = c.getPercent();
// receives the winner
int winner = c.getWinner();
// output
System.out.println("percent of voters is: "+percentVotters);
System.out.println("winner candidate is: "+winner);
}
}

```

פלט אפשרי:

```

Enter number of students
30
Enter vote, 3 student's number (1-20)
3
1
12
Enter vote, 3 student's number (1-20)
4
2
12
Enter vote, 3 student's number (1-20)
2
5
10
Enter vote, 3 student's number (1-20)
5
7
12
Enter vote, 3 student's number (1-20)
0
percent of voters is: 13.333333333333334
winner candidate is: 2

```

פתרון בסי שארפ – דורית בן דוד

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BagrutQ9
{
    class BagrutQ9
    {
        static void Main(string[] args)
        {
            int[] kelet = new int[] { 40, 10, 5, 15, 7, 15, 1, 8, 15, 4, 15, 20, 9, 1, 5, 8, 0 };

            int numStudents = kelet[0]; // int.Parse(Console.ReadLine());
            Competition competition = new Competition(numStudents);

            // קלט של שלשות עד...שייקלט 0
            int vote = kelet[1]; // int.Parse(Console.ReadLine());
            for (int i = 0; i < kelet.Length && vote != 0; i++) // while (vote!=0)
            {
                for (int point = 3; point >0 && vote!=0; point--)
                {
                    competition.AddPoints(vote, point);
                    vote = kelet[i * 3 + 2 + (3 - point)]; // int.Parse(Console.ReadLine());
                }
                competition.AddVotter();
            }
            Console.WriteLine("participate "+competition .GetPercent()+"% the winner is no.
                               "+competition.GetWinnerId ());
        }
    }
}

```

```

public class Competition
{
    int[] moamadim; // מערך צוברים למניית מספר הנקודות שקיבל כל מועמד
    int numStudents; // מספר כל תלמידי בית הספר
    int counter; // מספר התלמידים בהצבעה

    // בנאי
    //=====
    // טענת כניסה: מספר תלמידי בית הספר. גדול מאפס
    // טענת יציאה: נבנה עצם מסוג תחרות המייצר מערך צוברים בגודל מספר תלמידי בית הספר/
    // ומאפס את מספר התלמידים שהשתתפו בהצבעה.
    public Competition(int numStudents)
    {
        this.numStudents = numStudents;
        this.moamadim = new int[20 + 1]; // יש 20 מועמדים
        for (int i = 0; i < this.moamadim.Length; i++) // איפוס המערך
        {
            this.moamadim[i] = 0;
        }
        this.counter = 0;
    }
    // טענת כניסה : אין
    // טענת יציאה : התווספה הצבעת תלמיד עבור 0 מקומות ראשונים.
    public void AddVotter()
    {
        this.counter++;
    }
    // טענת כניסה : מספר זיהוי בין 1 ל 21 , הנקודות 1 או 2 או 3
    // טענת יציאה : התווספו הנקודות למועמד בעל מספר הזיהוי המתאים
    public void AddPoints(int id, int points)
    {
        this.moamadim[id] = this.moamadim[id] + points;
    }
    // טענת כניסה : אין.
    // טענת יציאה : הוחזר מספר האחוזים של התלמידים שהצביעו מסך הכל התלמידים.
    public double GetPercent()
    {
        return (double)this.counter / this.numStudents* 100;
    }
}

```

```

//טענת כניסה : אין//
// טענת יציאה : הוחזר מספרו הסידורי של התלמיד שקיבל את מירב הקולות (הנחה יש רק אחד)
// יש לשים לב שהתלמיד במיקום 1 במערך, מספרו הסידורי הוא 1
public int GetWinnerId()
{
    int imax = 0;
    for (int i = 0; i < this.moamadim.Length; i++)
    {
        if (this.moamadim[i] > this.moamadim[imax])
            imax = i;
    }
    return imax;
}
}
}

```

### שאלה 10 : Domino

#### הערות של EVI

- 1- בסעיף (א) כתוב ממש ואח"כ פירוט מה שורצים. בסעיף (ב) כתוב ממש פעולה מתוך אוסף פעולות הממשק. יש לדאוג לניסוח ברור. או לא לכתוב בסעיף (א) את המילה **ממש**. ולפרט בדיוק מה רוצים. או לכתוב איך תייצג את המחלקה. הכי טוב בעיני לרשום מה רוצים כלומר, רושם את כותרת המחלקה המייצגת "**אבן דומינו**", תכונות המחלקה, חתימות הפעולות כולל טענת כניסה ויציאה לכל פעולה.
- 2- בשה"כ שאלה טובה, מנוסחת טוב. ברורה.

#### הערות של דורית:

- א. השאלה מאולצת בהמרה לתוכנית החדשה.
- אין שימוש בפעולה isDouble...וגם לא נדרשו לממש אותה - אז...למה היא מופיעה בשאלה בכלל?
- ב. השאלה לא מוגדרת עד הסוף: מתאים זה ש  $ary[i].n1=ary[i+1].n2$  או הפוך ???
- ג. חסרה הנחה שהמערך דומינו הקיים הוא חוקי !!!
- ד. לא ברור ממש מה הכוונה ב-isFit ואם התלמיד כתב את התנאי הפוך אז...זה לא יעבוד!!
- ה. השאלה הזו נראית לי קלה מדי לרמה הזו, אפילו יותר מהשאלה הקודמת
- שאלתי שאלה דומה את התלמידים שלי במבחן
- אבל...אצלי בקוביית הדומינו לא היו שני מספרים שמתקבלים בבנאי אלא מספר שלם בודד וחיובי
- ו. isFit...זה אם ספרת האחדות מתאימה לספרה הגבוהה של קוביית הדומינו הקודמת.

למשל:

11, 12465, 5, 54, 4351, 123

זה נראה לי קצת יותר חביב ו...גם מתאים מבחינת הרמה ככ גם יותר ברור מי צריך להתאים למי

#### קשיים ושגיאות אפשריות של תלמידים

#### בתכנית הראשית:

- 1- בבדיקת התאמת זוגות אבני דומינו סמוכות, גלישה מגבולות המערך, כי ניגשים למערך במקום  $i+1$  או  $i-1$ .
- 2- אי ידיעה איך לזמן את הפעולה isFit במחלקת הבדיקה.
- 3- בבדיקת התאמת זוגות אבני דומינו סמוכות, טיפול לא נכון במתי "כן" ומתי "לא".

למשל, במקום להתחיל דגל לאמת ואם הבדיקה נכשלת לשנות לשקר, בכל שלב לשנות את ערך הדגל לאמת או שקר.

או, יותר גרוע, להדפיס "כן" או "לא" בכל שלב של הבדיקה במקום בתום הבדיקה.

-4 isFit בפעולה

-1 לא בצע קשר לוגי מתאים.

-2 לא להתייחס לכל המקרים, השוואה מספר ראשון לשני מספרי אבן שנייה וכו"ל למספר השני.

-3 אי ידיעה איך להשוות בין ערך מספר של האבן הנוכחית לאבן שמעברת כפרמטר.

פתרון בג'אווה:

```
/**
 * Question10:
 * @author evi
 */
public class Domino
{
    // properties: 2 points on stone
    private int n1;
    private int n2;
    /**
     * create Domino with points n1,n2
     * @param n1: int
     * @param n2 :int
     */
    public Domino(int n1, int n2)
    {
        this.n1 = n1;
        this.n2 = n2;
    }
    /**
     * @return first value on Domino
     */
    public int getN1()
    {
        return n1;
    }
    /**
     * @return second value on Domino
     */
    public int getN2()
    {
        return n2;
    }
    /**
     * @return true if 2 points are equal, false - otherwise
     */
    public boolean isDouble()
    {
        return this.n1==this.n2;
    }
}
```

```

/**
 * @param other: Domino
 * @return true if can link other to this Domino, false - otherwise
 */
public boolean isFit(Domino other)
{
    return this.n1==other.n1 || this.n1==other.n2 ||
           this.n2 == other.n1 || this.n2 == other.n2;
}
/**
 * @return String representing Domino
 */
public String toString()
{
    return "["+this.n1+", "+this.n2+"]";
}
}

import java.util.Random;
import java.util.Scanner;

/**
 * Question10:
 * @author evi

 * input: there is array of Domino in size 28
 * output: yes if it is "domino-snake", false-otherwise
 * "domino-snake" is sequence of dominos which can be linked together.
 */
public class RunDomino
{
    /**
     * create Domino[] with n stones
     * @param n: int, size of sequence of dominos
     * @return Domino[]
     */
    public static Domino[] createRandom(int n)
    {
        Random rnd = new Random();
        int n1,n2;
        Domino[] dominos = new Domino[n];
        for (int k=0; k<n; k++)
        {
            n1=rnd.nextInt(6)+1;
            n2=rnd.nextInt(6)+1;
            dominos[k]=new Domino(n1, n2);
        }
        return dominos;
    }
}

```

```

/**
 * create Domino[] with n stones
 * @param n: int, size of sequence of dominos
 * @return Domino[]
 */
public static Domino[] createInput(int n)
{
    Scanner in = new Scanner(System.in);
    int n1,n2;
    Domino[] dominos = new Domino[n];
    for (int k=0; k<n; k++)
    {
        System.out.println("Enter the 2 values on Domino(1-6)");
        n1 = in.nextInt();
        n2 = in.nextInt();
        dominos[k]=new Domino(n1, n2);
    }
    return dominos;
}
/**
 * prints dominos
 * @param d : Domino[]
 */
public static void printDominos(Domino[] d)
{
    for (int k=0; k<d.length; k++)
        System.out.print(d[k]+"-->");
    System.out.println("");
}
/**
 *
 * @param d : Domino[]
 * @return true if "snake-domino", false - otherwise
 */
public static boolean isSnake(Domino[] d)
{
    for (int k=1; k<d.length; k++)
    {
        if (!d[k].isFit(d[k-1]))
            return false;
    }
    return true;
}
public static void main(String[] args)
{
    Domino[] dominos = RunDomino.createInput(28);
    RunDomino.printDominos(dominos);
    boolean isSnake = RunDomino.isSnake(dominos);
    if (isSnake)
        System.out.println("yes");
    else
        System.out.println("no");
}

```

}

דוגמת פלט לנחש דומינו באורך 6

[1,5]--&gt;[5,3]--&gt;[3,6]--&gt;[6,2]--&gt;[2,2]--&gt;[2,4]--&gt;

**Yes**פתרון בסי שארפ:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BagrutQ10
{
    class BagrutQ10
    {
        static void Main(string[] args)
        {
            int[] kelet = new int[] { 1, 5, 5, 6, 6, 0, 0, 2, 2, 1 }; // שימוש בדוגמא שניתנה בשאלה
            Domino[] snake = new Domino[kelet.Length / 2]; // new Domino[28];

            // קלט למערך, הנחה: כבר בוצע
            /** מיותר בפתרון, רק לצורכי הרצה
                for (int i = 0; i < snake.Length; i++)
                {
                    snake[i] = new Domino(kelet[i * 2], kelet[i * 2 + 1]);
                }
            **/

            bool isSnake = true;
            for (int i = 0; i < snake.Length - 1 && isSnake; i++)
            {
                isSnake = snake[i].IsFit(snake[i + 1]);
            }
            Console.WriteLine("isSnake " + isSnake);
        }
    }

    public class Domino
    {
        int n1;
        int n2;
    }
}

```



```
// בנאי
//=====
//פעולה הבונה אבן דומינו בהתאם לפרמטרים
public Domino(int n1, int n2)
{
    this.n1 = n1;
    this.n2 = n2;
}
//פעולה המחזירה ערך המספר הראשון באבן
public int GetN1(){
    return this.n1;
}
//פעולה המחזירה ערך המספר השני באבן
public int GetN2()
{
    return this.n2;
}
// אם ערכי התכונות שווים זה לזה true פעולה המחזירה את,
//אחרת מחזירה false
public bool IsDouble()
{
    return this.n1 == this.n2;
}
// אם ניתן להצמיד את אבן הדומינו true פעולה המחזירה
//הנוכחית לאבן הדומינו שבפרמטר
public bool IsFit(Domino other)
{
    return this.n2 == other.n1;
}
}
}
```

### שאלות מתכונת שלמורי הניסוי

#### שא' 9 במתכונת:

- א. השאלה נראית לי קצת יותר ברמה המתאימה לחלק ג בבגרות אם כי לא נדרשת חשיבה וקוד בכתובת המחלקות רק תובנות של החומר בכתובת קוד התוכנית הראשית.
- ב. יש יותר מדי מאפיינים מיותרים ולא ברור מאיזה טיפוס מאפיין תאריך ( הטיפוס תאריך = DateTime לא נלמד בתוכנית הלימודים)  
והמאפיין זמן השיחה (int) זה קצת כי מדובר בשניות ו-long לא מחוייב בתוכנית הלימודים)
- ג. חסרה הנחה כי המערך מלא בצורה תקינה וכי הנחה שיש לפחות שיחה אחת במערך

#### בעיות שיכולות לצוץ:

- 1- גישה לפעולה שלא שייכת למחלקה הנכונה , כי יש 2 מחלקות בשימוש
- 2- אין איפוס/ איתחול של max ושל שם המשתמש
- 3- התנאי בבדיקה הפוך
- 4- בשיטה א: לא שומרים גם את שם המשתמש המקסימלי
- 5- לא מדפיסים בסוף את מה שצריך אלא מה שהשתמשנו בדרך..

פתרתי ב-2 צורות, אחת יותר מתאימה לתוכנית הישנה והשנייה יותר OO

#### פתרון השאלה בסי שארפ:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MatkonetQ9
{
    class MatkonetQ9
    {
        static void Main(string[] args)
        {
            Call[] calls = new Call[100]; // כבר בוצע קלט תקין למערך

            /** פתרון 1
            long max=0;
            User maxCaller=null;
            for (int i = 0; i < calls.Length; i++)
            {
                if (calls[i].GetDuration()>max )
                {
                    max=calls[i].GetDuration();
                    maxCaller =calls[i].GetCallerUser ();
                }
            }

            if (maxCaller!= null) // אם מערך השיחות לא ריק ו..נמצא כזה משתמש
                Console.WriteLine("max caller User name" + maxCaller .GetUserName());
            *
            **/
        }
    }
}
```

```

/** פתרון 2 */
Call max = calls[0]; // בהתחלה השיחה ה 1 הוא המקסימלית
for (int i = 1; i < calls.Length; i++)
{
    if (calls[i].GetDuration() > max.GetDuration() )
    {
        max=calls[i];
    }
}
Console.WriteLine("max caller User name" + max.GetCallerUser().GetUserName());
}
}

public class User //משתמש
{
    string userName; //שם משתמש
    int id; // מספר תעודת זהות
    string address; // כתובת
    int cellPhone; //מספר הטלפון של הטלפון הנייד

    // בנאי: הנחה, קיים
    public User(string userName,int id,string address,int cellPhone)
    {
        this.userName = userName;
        this.id = id;
        this.address = address;
        this.cellPhone = cellPhone;
    }

    // נדרש להוסיף
    // הפעולה מחזירה את שם המשתמש
    public string GetUserName()
    {
        return userName;
    }
}

public class Call //שיחה
{
    //משתמש מתקשר, משתמש מקבל השיחה, תאריך השיחה, משך זמן השיחה בשניות.
    User callerUser; //משתמש מתקשר
    User receiverUser; //משתמש מקבל השיחה
    DateTime date; // תאריך השיחה
    long /*int*/ duration; // משך זמן השיחה בשניות
}

```

```
// בנאי: הנחה, קיים
public Call(User callerUser, User receiverUser, DateTime date, long duration)
{
    this.callerUser = callerUser;
    this.receiverUser = receiverUser;
    this.date = date;
    this.duration = duration;
}

// נדרש להוסיף
// הפעולה מחזירה את אורך השיחה
public long GetDuration ()
{
    return duration;
}

// נדרש להוסיף
// הפעולה מחזירה את המשתמש שהתקשר
public User GetCallerUser ()
{
    return callerUser;
}
}
```

**שאלה 10 מהמתכונת:**

- א- חסר פה את התוכנית הראשית. כלומר לשם מה יש את המחלקות הללו בכלל?  
 ב- הרמה שוב קלה מדי לחלק ג, לדעתי  
 ג- לתשומת לב, ב #C לא חייבים את הפעולה equals להשוואת מחרוזות, אבל כתבתי ב-2 הצורות

בעיות שיכולות לצוץ:

- 1- אין פה יותר מדי קוד אבל...במה שיש  
 2- אפשר שלא לאפס את המונה  
 3- טעות בהשוואה, להשוות את הקטיגוריה ל- clip או אורך או לשם ...

פתרון בסי שארפ:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MatkonetQ10
{
    class MatkonetQ10
    {
        static void Main(string[] args)
        {
        }
    }

    public class Clip //קליפ
    {

        string name; //שם הקליפ
        string performer; //שם המבצע
        int length; //אורך הקליפ בשניות

        //בנאי
        public Clip(string name,string performer,int length)
        {
            this.name = name;
            this.performer = performer;
            this.length = length;
        }

        public string GetName()
        {
            return this.name ;
        }
        public string GetPerformer()
        {
            return this.performer;
        }
        public int GetLength()
        {
            return this.length;
        }
    }
}
```

```
// הפעולה מחזירה מחרוזת המתארת את אורך הקליפ הנוכחי כדלקמן:
//, "short" מחזירה הפעולה אחת, הפעולה מחזירה "long"
// אם אורכו מעל 3 דקות, הפעולה מחזירה "middle"
// . אחרת, הפעולה מחזירה "middle"
```

```
public string LengthCategory()
{
    if (this.length < 1 * 60)
        return "short";
    else
        if (this.length > 3 * 60)
            return "long";
        else
            return "middle";
}
```

```
// הפעולה מקבלת קליפ c
// ומחזירה true: אם המבצע של הקליפ שהועבר כפרמטר זהה למבצע הקליפ הנוכחי
// אחרת false.
```

```
public bool SamePerformer(Clip c)
{
    return this.performer == c.performer ;
    // return this.performer.Equals(c.performer);
}
}
```

```
public class ClipCollection // אוסף קליפים
{
    Clip[] Clips; // הנחה: המערך כבר מלא בנתונים
    string name; // שם האוסף

    // בנאי : הנחה, קיים
    public ClipCollection(string name)
    {
        this.name = name;
        //this.Clips; הנחה המערך כבר מלא בנתונים
    }

    public int CollectionLength()
    {
        return this.Clips.Length;
    }
}
```

```

// הפעולה מקבלת שם קטיגוריה
// ומחזירה את מספר הקליפים באוסף הקליפים
// אשר תיאור אורכם הוא הפרמטר category
public int CountLengthCategory(String category)
{
    int count = 0;
    for (int i = 0; i < this.Clips.Length; i++)
    {
        if (Clips[i].LengthCategory() == category)
            //if (Clips[i].LengthCategory().Equals (category))
                count++;
    }
    return count;
}
}
}

```

**שאלה 11: SuperInt**  
השלמת פעולות במחלקה.

הערות:

במימוש הפעולות האם-קעור? ו אחזר-מספר-מינימום-מקומי, יש כפילות קוד.  
לכן, אפשר לממש את הפעולה המחזירה את המינימום המקומי בתנאי שהתנאים למספר קעור מתקיימים, (-1) אם לא מספר קעור ובפעולה האם-קעור לזמן את החזרת מינימום-מקומי שונה מ (-1)

כלומר, הפעולה הבוליאנית האם מספר הוא קעור? מיותרת.  
בעזרת הערך המוחזר מהפעולה אחזר-מינימום-מקומי ניתן לדעת גם האם המספר הוא קעור או לא.  
ערך (-1) יציין שהמספר אינו קעור.  
כלומר כדאי לממש את הפעולה הבוליאנית "האם קעור?" תוך שימוש בפעולה "מינימום מקומי".

פתרונות:  
**אפשרות א:**

```

public boolean isConcave()
{
    // טענת יציאה: מחזיר אמת אם מספר-קעור, שקר-אחרת
    int num = this.theNumber;
    if (num < 10) // 1 digit is ok
        return true;
    // only 2 digits or not up at the end of the number
    if (num < 100 || num % 10 <= num / 10 % 10)
        return false;
    // check part of ascending
    while (num > 9 && num / 10 % 10 < num % 10)
        num = num / 10;
    if (num < 10) // only part of ascending
        return false;
    // check part of descending
    while (num > 9 && num / 10 % 10 > num % 10)
        num = num / 10;
    return num < 10;
}
}

```

```

public int localMin()
{// טענת יציאה: מחזיר את המינימום המקומי עבור מספר קעור, (-1) אם לא קעור
    if (!this.isConcave())
        return (-1);
    int num = this.theNumber;
    while (num/10%10<num%10)
        num = num/10;
    return num%10;
}

```

אפשרות ב:

```

public int localMin()
{// טענת יציאה: מחזיר את המינימום המקומי עבור מספר קעור, (-1) אם לא קעור
    int min;
    int num = this.theNumber;
    if (num<10)
        return num;
    if (num<100 || num/10%10>=num%10)
        return (-1);
    while (num>9 && num/10%10<num%10)
        num = num/10;
    if (num<10)
        return -1;
    min = num%10; // optionally to be local-minimum
    while (num>9 && num/10%10 >num%10)
        num = num/10;
    if (num>9)
        return -1;
    return min;
}

```

```

public boolean isConcave()
{// טענת יציאה: מחזיר אמת אם מספר-קעור, שקר-אחרת
    return this.localMin()!=-1;
}

```

בשתי האפשרויות :

```

public boolean isSameLocalMin(SuperInt other)
{// טענת כניסה: עצם מטיפוס סופר-מספר
    // טענת יציאה: אמת אם המספר והמספר האחר הם מספרים קעורים בעלי אותו מינימום מקומי, שקר-אחרת
    return this.localMin() != -1 &&
        this.localMin() == other.localMin();
}

```



**שאלה 12: Copepod**

עבודה עם טיפוס נתון, הוספת פעולות מחוץ למחלקה או קטעי קוד.

הערות:

- 1- סעיפים א', ב', ג' קלים.
- 2- סעיף ד' – קשה, מעל רמת יסודות מדעי המחשב.
- 3- היה נוח יותר לו הפעולות הנדרשות היו בתוך המחלקה

הערות של דורית:

- כפי שסעיף ד' של השאלה מנוסח, הסעיף לא פתיר:
- חסרים נתונים בשאלה כדי להגדיר "תת-קבוצה": (אותו מין אותו גיל ואותו אורך) :
  - המין זה זכר/נקבה כלומר 0/1 וזה בסדר
  - הגיל זה מספר בין 0 ל...לא כתוב !
  - האורך זה מספר בין 0.5 ל 3.0 מילימטרים ו...זה המון קטגוריות. אינסוף קטיגוריות, כי מדובר במספר ממשי.
- לטעמי יש לומר שהקופוד חי מקסימום 31 יום. (בדקתי באינטרנט)  
 וכן להגדיר את האורך בקטגוריות של כל 0.5 מילימטר – כלומר נקבל 5 קטיגוריות. (0.5 עד 1, 1 עד 1.5 וכיוצ"ב)  
 ו...אז זה פתיר !!!  
 אבל...מורכב מדי לרמת בחינת הבגרות, גם בחלק ג.

פתרון סעיף א: מימוש המחלקה Copepod

```
/**
 * @version 23/2/14
 * @author evi
 */
public class Copepod
{
    private double length; // length of copepod
    private int gender;    // the gender, 1 - female , 0 - male
    private int age;      // the age

    public Copepod(double length, int gender, int age)
    {
        this.length = length;
        this.gender = gender;
        this.age = age;
    }

    public double getLength()
    {
        return length;
    }

    public int getAge()
    {
        return age;
    }
}
```

```

public int getGender()
{
    return gender;
}

@Override
public String toString()
{
    String str="Copepod:";
    str+="age:"+this.age+", gender:";
    if (this.gender==0)
        str+="male";
    else
        str+="female";
    str+=", length:"+this.length;
    return str;
}
}

```

```

Copepod c = new Copepod(0.35, 0, 1);
System.out.println(c);

```

פתרון סעיף ב:פתרון סעיף ג:

```

/**
 * (ג) prints length of oldest copepod
 * @param copepod : Copepod[]
 */
public static void lengthOldest(Copepod[] copepod)
{
    int maxAge=copepod[0].getAge();
    int indexOldest = 0;
    for (int k=1; k<copepod.length; k++)
        if (copepod[k].getAge(>maxAge)
            {
                maxAge = copepod[k].getAge();
                indexOldest = k;
            }
    System.out.println("length of oldest is:"+copepod[indexOldest].getLength());
}

```

זימון הפעולה הראשית:

```

Copepod[] copepodArr = RunCopecod.getAndCreate(5);
RunCopecod.lengthOldest(copepodArr);

```

פתרון סעיף ד:**משתנים נדרשים:**

1. מערך עזר arr1 מטיפוס Copepod , כדי לא להרוס את המערך המקורי
2. מערך arrSub מטיפוס Copepod אשר ישמור את הקופודים השונים זה מזה.
3. מערך counter אשר ישמור כמה פריטים יש מכל לל-קבוצה
4. מספר שלם, count אשר ימנה את מספר הפריטים בתת-הקבוצה הנוכחית
5. מספר שלם duffer אשר ימנה את מספר תתי-הקבוצות

**רעיון הפתרון:**

1. ראשית נעתיק את אברי מערך מקורי למערך עזר כדי לא לקלקל את המערך המקורי.
2. ניצור מערך שישמור את תתי הקבוצות של הקופודים ומערך שישמור את מספר הפריטים מכל תת-קבוצה.
3. נעבור על מערך העזר, עבור על איבר, נעתיק אותו למערך תת-הקבוצות ונאתחל את מספר הפריטים ל 1
4. מהמקום הבא במערך נעזור ונשווה לאיבר שזה עתה הצבנו במערך תת-הקבוצות. במידה ופרטיו זהים, נגדיל את מונה מספר הפריטים ב 1 ונציב null בתא זה במערך.
5. נמשיך כך עד למעבר על כל תאי המערך.
6. בסיום נציב את מונה מספר הפריטים במערך המנייה. נגדיל את ערך משתנה הסופר את מספר תתי-הקבוצות.
7. נחזור על סעיפים 3-6

```

public static void subClasses(Copepod[] arr)
{
    // copy the array for not destroying original array
    Copepod[] arr1 = new Copepod[arr.length];
    for (int k=0; k<arr.length; k++)
        arr1[k]= arr[k];
    // array with sub-classes
    Copepod[] arrSub = new Copepod[arr.length];
    // array with number of items in each sub-class
    int[] counter = new int[arr.length];
    int count;
    int differ=0; // number of sub-classes
    int index1=0;
    int index2=0;
    while (index1<arr1.length)
    { // starts sub-category
        if (arr1[index1]!=null)
        {
            arrSub[differ] = arr1[index1];
            count = 1;
            index2 = index1+1;
            while (index2<arr1.length)
            {
                if (arr1[index2]!=null && RunCopecod.isSame(arrSub[differ], arr1[index2]))
                {
                    count++;
                    arr1[index2]=null;
                }
                index2++;
            }
            counter[differ]=count;
            differ++;
        }
        index1++;
    }
    System.out.println("number of sub-classes is: "+differ);
}

```

```

for (int k=0; k<differ; k++)
{
    System.out.println("number of items is sub-class is: " + counter[k]);
    System.out.println(arrSub[k]);
    System.out.println("-----");
}

```

זימון בפעולה הראשית:

RunCopecod.**subClasses**(copepodArr);

דוגמה לפלט:

```

run:
Compepod:age:1, gender:male, length:0.35
Enter length, gender(0-male, 1-female) , age
0.2
1
2
Enter length, gender(0-male, 1-female) , age
0.3
0
1
Enter length, gender(0-male, 1-female) , age
0.2
1
2
Enter length, gender(0-male, 1-female) , age
0.3
0
3
Enter length, gender(0-male, 1-female) , age
0.3
0
1
length of oldest is:0.3
number of sub-classes is: 3
number of items is sub-class is: 2
Compepod:age:2, gender:female, length:0.2
-----
number of items is sub-class is: 2
Compepod:age:1, gender:male, length:0.3
-----
number of items is sub-class is: 1
Compepod:age:3, gender:male, length:0.3

```