

```
1: public class Hello {
2:     public static void main(String[] args) {
3:         System.out.println("*****");
4:         System.out.println("*****Welcome to Java!*****");
5:         System.out.println("*****");
6:     }
7: }
```

לאחר הרצת התכנית יופיע בחלון output הפלט הבא:

```
run:
*****
****Welcome to Java!****
*****
```

תרגיל כתי

כתבו תכנית המציגה בחלון הפלט, בשורה אחת, את שמכם ובשורות הבאות פרטים על עצמכם (למשל מקום מגורים, גיל, צבע מועדף וכו').

```
1: import java.util.Scanner; // import class Scanner
2: public class SayHello {
3:     public static void main(String[] args) {
4:         // create object to read input from screen
5:         Scanner input = new Scanner(System.in);
6:         String name;
7:         // ask for user's name
8:         System.out.println("What is your name?");
9:         name = input.next(); // read in user's name
10:        System.out.println("Hello "+name);
11:    }
12: }
```

לאחר ביצוע שורה 8 יופיע בחלון הפלט הטקסט: What is your name?
בעת ביצוע שורה 9, יפתח חלון הinput בתחתית המסך והתכנית תעצר ותמתין. המשתמש יכתוב את הנדרש בחלון הקלט והתכנית תמשיך רק לאחר שהמשתמש יקיש על מקש ה - enter.

תרגיל 1

כתבו תכנית המבקשת המשתמש להקליד את מספר הטלפון שלו. המספר ייקלט כטקסט. התכנית תוסיף לתחילת המספר את הקידומת 9 ותוציא הודעה המציגה למשתמש את המספר עם הקידומת ותסביר למשתמש שזהו מספרו החדש. עליכם להגיש הדפסה של התכנית ושל הקלט והפלט באופן הבא: העתיקו את הקלט והפלט לחלון התכנית כך שיופיע אחרי התכנית והדפסו על ידי בחירה בתפריט file בפקודה print. לאחר ההדפסה, מחקו מהחלון את הקלט והפלט כך שתישמר רק התכנית.

```

1: import java.util.Scanner;
2: public class add2Nums {
3:     public static void main( String args[] ){
4:
5:         Scanner readIn = new Scanner(System.in);
6:         int num1, num2,    // the 2 numbers we want to add
7:         sum;              // the sum of the 2 numbers
8:         // instruct user to enter 2 numbers
9:         System.out.println("Enter 2 numbers a space between them?");
10:        num1 = readIn.nextInt(); // read in frist num
11:        num2 = readIn.nextInt(); // read in second num
12:        sum = num1 + num2;      // add the numbers
13:        System.out.println("The sum is "+ sum);
14:    }
15: }

```

```

1: import java.util.Scanner;
2: public class PhoneCall {
3:     // Calculate the time a phone call will end given: start time and duration of call.
4:
5:     public static void main(String[] args) {
6:         Scanner readIn = new Scanner(System.in);
7:         int hours, minutes, seconds, duration;
8:         System.out.println("Enter start time as 3 integers, hours, minutes, seconds");
9:         // Read in star time
10:        hours = readIn.nextInt(); // read in hours
11:        minutes =readIn.nextInt(); // read in minutes
12:        seconds = readIn.nextInt(); // read in seconds
13:        System.out.println("Enter duration");
14:        duration = readIn.nextInt(); // read in duration
15:        // calculate time of end of conversation
16:        seconds = seconds + duration;
17:        minutes = minutes + seconds / 60;
18:        seconds = seconds % 60;
19:        hours = hours + minutes / 60;
20:        minutes = minutes % 60;
21:        hours = hours % 24;
22:        // Display result
23:        System.out.println("End of conversation at " + hours + ":" + minutes + ":" +
24:            seconds);
25:    }

```

תרגיל 2

כתבו תכנית הקולטת מהמשתמש מספר תלת-ספרתי ומציבה אותו במשתנה מספרי שלם (אחריות המשתמש לספק מספר העונה על הדרישות, התכנית יכולה להניח קלט תקין) התכנית תציב במשתנה מספרי אחר את המספר הנוצר מהיפוך הספרות של המספר שנקלט ותציג אותו על המסך. דוגמה: אם נקלט המספר 135 יש להציג על המסך את המספר 531. בזמן ההרצה, אל תכתבו כקלט, מספר המסתיים באפס

אופן הדפסה:

העתיקו את הפלט לחלון התכנית, מיד אחרי התכנית. הדפיסו ע"י בחירה בתפריט FILE בפקודה PRINT.

DEBUGGING

נוח לפתור שגיאות לוגיות ע"י הצגת תוכן משתנים בשלבים שונים של ההרצה. מעקב אחר תוכנם עשוי לעזור בפתרון הבעיה הלוגית. עשו זאת בעזרת ההוראה

```
System.out.println ( );  
למשל כדי לעקוב אחר תוכן המשתנה b ניתן להוסיף במקומות שונים בתכנית את ההוראה:  
System.out.println ( " b = " + b );  
יש להסיר הוראות אלו לפני ההגשה.
```

```
-----  
1: // Read student's mark in test, add 10% to marks below 70  
2: // add 5% to marks 70 and above  
3: import java.util.Scanner;  
4: public class IfExample {  
5:     public static void main(String[] args) {  
6:         Scanner myIn = new Scanner(System.in);  
7:         double mark, newMark;  
8:         System.out.println("Enter your mark please");  
9:         mark=myIn.nextDouble();  
10:        // determine final mark  
11:        if (mark <70 ){  
12:            newMark = mark*1.1;  
13:            System.out.println("10% was added \nYour new mark is:"+newMark );  
14:        }  
15:        else{  
16:            newMark = mark*1.05;  
17:            System.out.println("5% was added \nYour new mark is:"+newMark );  
18:        }  
19:    }  
20: }
```

```

1: import java.util.Scanner;
2: public class Grade {
3: // Read student's mark in test,
4: // determine student's grade
5: public static void main(String[] args) {
6:     Scanner myIn = new Scanner(System.in);
7:     String ans; // answer to user
8:     int mark ; // mark of student
9:     System.out.println("Enter your mark please");
10:    mark=myIn.nextInt();
11:    ans="";
12:    // determine grade
13:    if (mark < 0 || mark >100 )
14:        ans="incorrect";
15:    else
16:        if (mark < 50 && mark >= 0)
17:            ans="fail";
18:        else if ( mark <=100 && mark >=50)
19:            ans="pass";
20:    // display the results
21:    System.out.println( "your grade is " + ans);
22: }
23: }

```

תרגיל 3

כתבו תכנית הקולטת שלושה מספרים שלמים. השלשה אמורה לייצג צלעות של משולש. עליכם להדפיס אחת מן המסקנות הבאות:

- א. המספרים אינם יכולים לייצג צלעות משולש מפני שאינם מקיימים את ההגדרה: סכום כל זוג צלעות במשולש גדול מן הצלע הנותרת. (דוגמה: השלשה 2,3,20 אינה מקיימת את ההגדרה משום ש $2+3$ אינו גדול מ 20)
 - ב. המספרים מייצגים צלעות משולש שווה צלעות (דוגמה 4, 4, 4)
 - ג. המספרים מייצגים צלעות משולש שווה שוקיים (דוגמה 5, 5, 4)
 - ד. המספרים מייצגים צלעות של משולש רגיל (שאינו שווה צלעות ואינו שווה שוקיים) (דוגמה 4, 8, 5)
- השתמשו בהוראות IF מקוננות ואל תבצעו בדיקות תקינות.

שימו לב : כתבו את התכנית כך שתודפס רק המסקנה הנכונה ביותר לגבי השלשה, כלומר, אם המשולש הוא שווה צלעות יש להציג הודעה מתאימה למרות שמשולש כזה יענה גם על ההגדרה של משולש שווה שוקיים.

** בדקו שהתכנית נותנת תשובה נכונה עבור כל אחת מארבע האפשרויות הנ"ל.

הציגו את הקלט והפלט (כפי שהוסבר בתרגילים קודמים) עבור אחת מהשלשות.

```

1: import java.util.Scanner; // import class Scanner
2: public class WhileExample {
3:     public static void main(String[] args) {
4:         Scanner input = new Scanner(System.in);
5:         String ans;
6:         int times, i;
7:         // Display "times" stars;
8:         System.out.println("How many stars would you like?");
9:         times=input.nextInt();
10:        ans=" Here are "+times+" stars: \n";
11:        i=1;
12:        while (i<=times){
13:            ans=ans+"*";
14:            i++; // i++ is the same as i=i+1
15:        }
16:        System.out.println(ans);
17:
18:        // Display numbers from 1 to times;
19:        ans=" Numbers from 1 to "+times+" are: \n";
20:        i=1;
21:        while (i<=times ){
22:            ans=ans+i+" ";
23:            i++;
24:        }
25:        System.out.println(ans);
26:        // Display even numbers from 2 to 20;
27:        // ans=" Even numbers from 2 to 20 are: \n";
28:        // i=
29:        // while (      ){
30:        //
31:        //
32:        // }
33:        // System.out.println(ans);
34:    }
35: }

```

```

//compute average of numbers entered by user
1:  import java.util.Scanner;
2:  public class Average {
3:      public static void main(String[] args) {
4:          Scanner input = new Scanner(System.in);
5:          int howMany,      // number of numbers to be entered
6:          sum,              // sum of numbers
7:          counter,         // counter of numbers
8:          number;          // number entered
9:          double average;  // average of all numbers
10:         // Initialization
11:         sum = 0;
12:         counter = 1;
13:         System.out.println("Enter number of numbers");
14:         howMany=input.nextInt();
15:         // Processing
16:         while ( counter <= howMany ) { // loop "howMany" times
17:             // read number from user as String and convert to type int
18:             System.out.println("number please");
19:             number = input.nextInt();
20:             // add number to sum
21:             sum = sum + number;          // shortcut: sum +=number;
22:             // add 1 to counter
23:             counter = counter + 1;      // shortcut: counter++;
24:         } // end of loop
25:         // Termination
26:         average = (double) sum / howMany; // perform real division
27:         // display average of numbers
28:         System.out.println( "Average is " + average );
29:     }
30: }

```

```

1: //compute average of numbers entered by user
2: import java.util.Scanner;
3: public class Average1 {
4:     public static void main(String[] args) {
5:         Scanner input = new Scanner(System.in);
6:         int sum,          // sum of numbers
7:         counter,        // counter of numbers
8:         number,         // number entered
9:         max ;           // maximum number
10:        double average; // average of all numbers
11:        // Initialization
12:        sum = 0;
13:        counter = 0;
14:        // Processing\
15:        // Read first number from user
16:        System.out.println("Enter Integer, -1 to Quit");
17:        number = input.nextInt();
18:        max = number;
19:        while ( number != -1 ) {
20:            sum = sum + number;    // add number to sum
21:            counter = counter + 1; // add 1 to counter
22:            if ( number > max )    // check if greater then max
23:                max = number;
24:            // Read next number from user
25:            number = input.nextInt();
26:        }
27:        // Termination
28:        if ( counter != 0 ) {
29:            average = (double) sum / counter;
30:            // display average and max of numbers
31:            System.out.println("Average is " + average
32:                               + "\n Maximum is " + max );
33:        }
34:        else
35:            System.out.println("No numbers were entered" );
36:    }
37: }

```

```

1: // Divide 2 whole numbers using subtraction and addition only
2: import java.util.Scanner; // import class Scanner
3: public class Div2Nums {
4:     public static void main(String[] args) {
5:         Scanner input = new Scanner(System.in);
6:         int num, div, count, remain;
7:         // read number and divider
8:         System.out.println("Please enter whole number and divider");
9:         num= input.nextInt();
10:        div=input.nextInt();
11:        // calculate num divided by div
12:        count=0;
13:        remain=num;
14:        while (remain>=div){
15:            count++;
16:            remain=remain-div;
17:        }
18:        System.out.println( num+" divided by "+div+" is "+count+" remainder "+remain);
19:    }
20: }

```

```

1:     import java.util.Scanner;
2: public class GuessNum {
3:     public static void main(String[] args) {
4:         Scanner input = new Scanner(System.in);
5:         int number, guess;
6:         String msg;
7:         number=(int)(101*Math.random()); // computer "selects" number
8:         // read first guess from user
9:         msg="Let's play! Guess my number (between 0 and 100)";
10:        do {
11:            System.out.println(msg);
12:            guess= input.nextInt();
13:            if (guess < number)
14:                msg="My number is bigger, try again";
15:            else
16:                if (guess > number)
17:                    msg="My number is smaller, try again";
18:        } while (guess != number);
19:        System.out.println( "You guessed it!!, "+number );
20:    }
21: }

```

תרגיל 4

כתבו תכנית הקולטת מספרים חיוביים שלמים ומציגה את המספר הקטן ביותר שנקלט (המינימום) ואת המספר השני הקטן ביותר. מספר המספרים אינו ידוע. רשימת המספרים תסתיים במספר שלילי. הריצו את התכנית לפחות שתי פעמים ובדקו שהתוצאות שקיבלתם נכונות. בהרצה הראשונה השתמשו בנתונים

הבאים:

12
5
30
11
3
37
8
-3

בהרצה השניה השתמשו בנתונים הבאים:

2
30
8
-3

הציגו כפלט את התוצאות המתקבלות משתי הרצות אלו.

אם הדבר מקל עליכם, מותר לכם להניח שהמשתמש יכניס לפחות 3 מספרים ושכל מספר יכול להופיע רק פעם אחת.

אין לבצע בדיקות תקינות!

```
1: import java.util.Scanner;
2: public class ConstantsEx {
3:     public static void main(String[] args) {
4:         Scanner myRead = new Scanner(System.in);
5:         final int FROM_KG = 40, TO_KG = 100;
6:         final double RATIO= 2.20462;
7:         int kg = FROM_KG;
8:         double pound;
9:         System.out.println( " KILOGRAM\tPOUNDS" );
10:        while ( kg <= TO_KG ) {
11:            pound = kg * RATIO; // convert kg to pound
12:            System.out.println(" " +kg +"\t\t" + pound );
13:            kg = kg + 5; // increment the kg value
14:        }
15:    }
16: }
```

```

2: public class ForExample{
3:   public static void main(String[] args){
4:     String s;
5:     int i;
6:     final int NUM = 10;
7:     // Display NUM stars;
8:     s=" Here are "+NUM+" stars: \n";
9:     for (i=1;i<=NUM;i++)
10:      s=s+"*";
11:     System.out.println( s);
12:     // Display numbers from 1 to NUM;
13:     s=" Numbers from 1 to "+NUM+" are: \n";
14:     for (i=1;i<=NUM;i++)
15:      s=s+i+" ";
16:     System.out.println( s);
17:     // Display even numbers from 2 to 2XNUM;
18:     s=" Even numbers from 2 to "+2*NUM+" are: \n";
19:     for (i=2;i<=2*NUM;i=i+2)
20:      s=s+i+" ";
21:     System.out.println( s);
22:   }
23: }

```

תרגיל 5 (הדוגמה הבאה Perfect While נכללת בחומר של שיעור 5)

כתבו תכנית הקולטת 10 מספרים ומחשבת ומציגה כפלט:

1. רק מבין 5 המספרים הראשונים, כמה מתחלקים ב-7 ללא שארית

2. מה אחוז המספרים החיוביים

3. מהו הערך המקסימלי מבין הערכים השליליים בלבד. הניחו שהמספר הקטן ביותר שיכול להתקבל הוא -10000.

עבור הקלט הבא:

17
-89
34
-22
14
90
77
-90
10
9

יהיה הפלט:


Among the first 5 numbers, 1 divides by 7
70%.are positive

-22 is the max of the negative numbers

```
1: // Check for perfect numbers
2:  import java.util.Scanner;
3: public class PerfectWhile{
4:  public static void main(String[] args){
5:    Scanner myRead = new Scanner(System.in);
6:    int number,sum,i;
7:    // read the number
8:    System.out.println(" Enter number please ");
9:    number = myRead.nextInt();
10:   // Number is perfect if sum of its dividers
11:   // (including 1 and not including the number itself) equals the number.
12:    sum=0;
```

```
13:    i=1;
14:    while (i<number){
15:      if ((number%i) == 0 )
16:        sum = sum + i;
17:      i++;
18:    } // end of while
```

```
19:    if (sum == number)
20:      System.out.println(number + " is perfect" );
21:    else
22:      System.out.println(number + " is not perfect");
23:  }
24: }
```



```
for (i=1; i<number;i++)
    if ((number%i) == 0 )
        sum = sum + i; // end of for
```

```

1: // read numbers from user until a negative number is entered
2: // or until 10 numbers have been read
3: import java.util.Scanner;
4: public class UseBoolean {
5:     public static void main(String[] args) {
6:         Scanner readIn = new Scanner(System.in);
7:         int num;
8:         boolean negative = false;
9:         for (int i = 1; i <= 10 && negative == false; i++){
10:             System.out.println("Enter a number please");
11:             num = readIn.nextInt();
12:             if (num < 0)
13:                 negative = true;
14:         }
15:         //Did user enter a negative number?
16:         if (negative == true)
17:             System.out.println("Loop ended because a negative num was encountered");
18:         else
19:             System.out.println("Loop ended because 10 nums were read");
20:     }
21: }

```

```

for (int i = 1; i <= 10 ; i++){
    System.out.println("Enter a number please");
    num = readIn.nextInt();
    if (num < 0){
        negative = true;
        break;
    }
}

```

```

1: // create multiplication table of 10 by 10
2: package kefel;
3: public class Kefel{
4:     public static void main(String[] args){
5:
6:         for (int i=1;i<=10;i++){
7:             for (int j=1; j<=10; j++)
8:                 System.out.print (i*j+"\t");
9:             System.out.println();
10:        }
11:    }
12: }

```

תרגיל 6

בחיבת ביניים מסוימת יש 3 שכבות. בכל שכבה יש 4 כיתות. עליכם לטפל ב- 12 מספרים המיצגים את מספר התלמידים בכל כיתה ולהציג את מספר התלמידים בכל אחת משלוש השכבות וכן את מספר התלמידים בכל בית הספר.

המספרים יתקבלו בסדר הבא: 4 המספרים המיצגים את מספר התלמידים בכל אחת מ-4 הכיתות של השכבה הראשונה, אח"כ 4 המספרים המיצגים את מספר התלמידים בכל אחת מ-4 הכיתות של השכבה השנייה ולבסוף 4 המספרים המיצגים את מספר התלמידים בכל אחת מ-4 הכיתות של השכבה השלישית. במקום לקלוט את 12 המספרים מהמשתמש, הגרילו 12 מספרים בין 20 ל-40.

לדוגמה, אם המספרים שהוגרלו הם:

20, 20, 35, 35, 40, 20, 21, 40, 35, 33, 27, 25

התכנית תציג:

Sum for grade 7 :110

Sum for grade 8 :121

Sum for grade 9 :120

Sum for school : 351

```

// using JOptionPane class to read and write
1: import javax.swing.*;
2: public class ReadAndWrite {
3:     public static void main(String[] args) {
4:         JOptionPane.showMessageDialog(null,"Messages can be written this way too");
5:         String s;
6:         int age;
7:         double heightM, heightI;
8:         s = JOptionPane.showInputDialog("Your name please");
9:         JOptionPane.showMessageDialog(null,"Hello "+s);
10:        s = JOptionPane.showInputDialog("Your age please");
11:        age=Integer.parseInt(s);
12:        JOptionPane.showMessageDialog(null,"You were born in "+(2005-age));
13:        s = JOptionPane.showInputDialog("Please enter your height in meters");
14:        heightM=Double.parseDouble(s);
15:        heightI = (heightM*100) * 0.03281;
16:        JOptionPane.showMessageDialog(null,"Your height is "+heightI+" feet");
17:    }
18: }

```

```

1: // Recieve number between 1 and 5 and write the number as a word
2:     import javax.swing.*;
3: public class SwitchEx {
4: public static void main(String[] args) {
5:     String sChoice;
6:     int choice;
7:     sChoice= JOptionPane.showInputDialog ("Write a number between 1 and 5");
8:     choice = Integer.parseInt(sChoice);
9:     switch (choice) {
10:         case 1: JOptionPane.showMessageDialog(null,"one"); break;
11:         case 2: JOptionPane.showMessageDialog(null,"two"); break;
12:         case 3: JOptionPane.showMessageDialog(null,"three"); break;
13:         case 4: JOptionPane.showMessageDialog(null,"four"); break;
14:         case 5: JOptionPane.showMessageDialog(null,"five"); break;
15:         default: JOptionPane.showMessageDialog(null,"Not a valid choice");
16:     } // end of switch
17: }
18: }

```

שיטות של המחלקה String (רשימה חלקית)
לצורך הדגמה נניח שהוגדרה ואתחלה המחזורת הבאה:

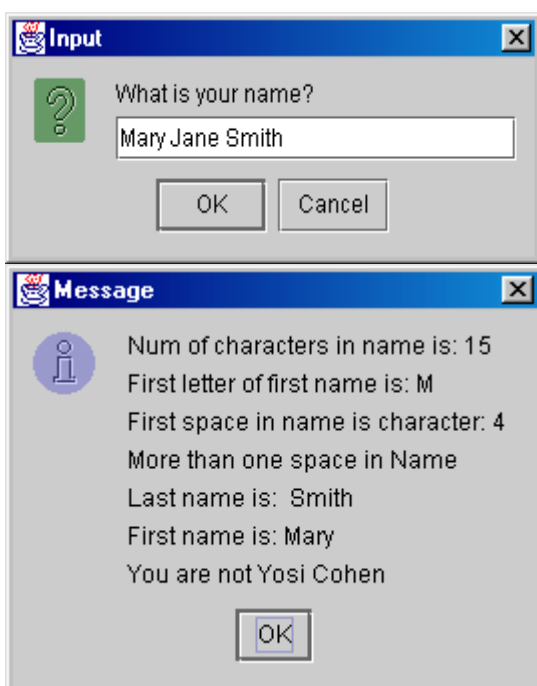
String words ;
words = "This is an example" ;

- (1) length() - מחזירה את אורך המחזורת (מספר התווים).
דוגמה: `int x= words.length();` יציב ב X את המספר 18
- (2) charAt(int index) - מקבלת מספר שלם index ומחזירה את התו הנמצא במקום ה - index של המחזורת. (האינדקס של המקום הראשון הוא 0).
דוגמה: `char c = words.charAt(1);` תציב את הערך 'h' במשתנה c.
- (3) indexOf(char ch) - מקבלת תו ch ומחזירה את האינדקס של המופע הראשון של התו במחזורת. אם התו לא נמצא במחזורת - מחזירה -1.
דוגמה: `int j = words.indexOf('s');` תציב את הערך 3 במשתנה j.
- (4) lastIndexOf(char ch) - מקבלת תו ch ומחזירה את האינדקס של המופע האחרון של התו במחזורת.
דוגמה: `int j = words.lastIndexOf('s');` תציב את הערך 6 במשתנה j.
- (5) substring(int index1, int index2) - מקבלת 2 אינדקסים ומחזירה אובייקט מסוג String המכיל תת-מחזורת של המחזורת באובייקט הנוכחי - החל מהתו במקום index1, עד לתו במקום index2-1.
דוגמא: `String s1 = words.substring(5, 7);` תציב ב s1 את המחזורת "is"
- (6) substring(int index) מקבלת אינדקס ומחזירה אובייקט מסוג String המכיל תת-מחזורת של המחזורת באובייקט הנוכחי - החל מהתו במקום index ועד לסוף המחזורת.
דוגמא: `String s1 = words.substring(11);` תציב ב s1 את המחזורת "example"
- (7) equals(String s) - מקבלת מחזורת ומחזירה את הערך true אם האובייקט והפרמטר מכילים את אותה המחזורת.
שימו לב שההוראה: `if(s1 == s2)` מחזירה את הערך true אם s1 ו s2 מתייחסים לאותו אובייקט.
דוגמא: לבדיקה אם המחזורת s מכילה "yes" נכתוב: `if(s.equals("yes"))`
- (8) equalsIgnoreCase(String s) מקבלת מחזורת ומחזירה את הערך true אם האובייקט והפרמטר זהים פרט ל case (גודל האותיות).
דוגמא: לבדיקה אם המחזורת s מכילה "yes" או "YES" או "Yes" נכתוב: `if(s.equalsIgnoreCase("yes"))`

```

1: //Using String methods
2:     import javax.swing.JOptionPane;
3: public class PlayWithStrings {
4:     public static void main(String[] args) {
5:         String ans = new String("");
6:         String fullName= JOptionPane.showInputDialog("What is your name?");
7:         int len = fullName.length();
8:         ans="Num of characters in name is: "+len+" \n";
9:         char first = fullName.charAt(0);
10:        ans = ans + "First letter of first name is: "+first+" \n";
11:        int firstSpace = fullName.indexOf(' ');
12:        ans = ans + "First space in name is character: "+firstSpace+" \n";
13:        int lastSpace = fullName.lastIndexOf(' ');
14:        if (firstSpace == lastSpace)
15:            ans = ans + "Only one space in Name \n";
16:        else
17:            ans = ans + "More than one space in Name \n";
18:        String lastName = fullName.substring(lastSpace);
19:        ans = ans + "Last name is: " + lastName + " \n";
20:        String firstName = fullName.substring(0,firstSpace);
21:        ans = ans + "First name is: " + firstName+ " \n";
22:        if (fullName.equals("Yosi Cohen"))
23:            ans = ans + "Hi Yosi Cohen \n";
24:        else
25:            ans = ans + "You are not Yosi Cohen \n";
26:        JOptionPane.showMessageDialog(null,ans);
28:    }
29:}

```



תרגיל 7

כתבו תכנית הקולטת 2 מספרים שלמים חיוביים גדולים משתיים. המספרים מגדירים טווח. התכנית תציג על המסך את כל המספרים הראשוניים בטווח.

לדוגמה: אם נקלטו המספרים 6 ו 24 יוצגו על המסך המספרים 7,11,13,17,19,23
אם נקלטו המספרים 25 ו 28 תופיע הודעה האומרת שאין בטווח זה מספרים ראשוניים.

מספר ראשוני הוא מספר המתחלק, ללא שארית, רק באחד ובעצמו.

הניחו קלט תקין, כלומר הניחו שקיבלתם שני מספרים חיוביים שלמים גדולים מ2 ושהמספר הראשון

שנקלט קטן מהמספר השני. אל תבצעו בדיקות תקינות!

אם אתם משתמשים ב-JOptionPane לקריאה ולכתיבה, הקלטים והפלטים מודפסים ע"י לחיצה על ALT + PRINT SCREEN כאשר החלון המבוקש מופיע, הדבקה (PASTE) בחלון PAINT והדפסת החלון PAINT.

```
1: import javax.swing.JOptionPane;
2: public class Menu {
3:     public static void main(String[] args) {
4:         String sChoice, sAns, s;
5:         int choice, num, ans;
6:         boolean tryGuess;
7:         tryGuess=false;
8:         do {
9:             sChoice= JOptionPane.showInputDialog
10:             ("Please choose: 1=Guess num 2=Multiplication question 3=favorite animal 4=Quit");
11:             choice = Integer.parseInt(sChoice);
12:             switch (choice){
13:                 case 1:
14:                     tryGuess=true;
15:                     sAns= JOptionPane.showInputDialog ("Guess my num (between 0 and 3)");
16:                     ans = Integer.parseInt(sAns);
17:                     num=(int)(Math.random()*4);
18:                     if (ans==num)
19:                         JOptionPane.showMessageDialog(null,"You guessed it!");
20:                     else
21:                         JOptionPane.showMessageDialog(null,"No, it's "+num);
22:                     break;
```

```

23:     case 2:
24:         int a=(int)(Math.random()*10);
25:         int b=(int)(Math.random()*10);
26:         sAns= JOptionPane.showInputDialog ("How much is "+a+" X "+b+"?");
27:         ans = Integer.parseInt(sAns);
28:         if (ans==a*b)
29:             JOptionPane.showMessageDialog(null,"correct!");
30:         else
31:             JOptionPane.showMessageDialog(null,"No, it's "+a*b);
32:         break;
33:     case 3:
34:         sAns = JOptionPane.showInputDialog("What is your favorite animal?");
35:         if (sAns.equals("dog"))
36:             JOptionPane.showMessageDialog(null,"Hey, we like the same animal!");
37:         else
38:             JOptionPane.showMessageDialog(null,"That's nice!");
39:         break;
40:     case 4:
41:         if (tryGuess == true)
42:             JOptionPane.showMessageDialog(null,"BYE");
43:         else
44:             JOptionPane.showMessageDialog(null,"Next time try to guess my num");
45:         break;
46:     default: JOptionPane.showMessageDialog(null,"Not a valid choice");
47: } // end of switch
48: } // end of do
49: while (choice !=4);
50: }
51: }

```

```
1: public class Methods1{
2:   public static void main(String[] args){
3:     // Using a method to print stars
4:     System.out.println( "I will print stars");
→ 5:     stars();
→ 6:     System.out.println( "and again stars");
→ 7:     stars();
8:   }
9:   public static void stars()
10:  {
11:    System.out.println("***");
12:  }
13: }
```

```
1:   import java.util.Scanner;
2: public class Methods2{
3:   public static void main(String[] args){
4:     Scanner input = new Scanner(System.in);
5:     int num;
6:     // Using a method to print the number of stars user wants
7:     System.out.println( "Please enter number of stars" );
8:     num=input.nextInt();
9:     stars(num);
10:    System.out.println("I will print 3 stars");
11:    stars(3);
12:   } // of method main
13:   public static void stars(int num){ ←
14:     String s="";
15:     for (int i = 1;i<=num;i++)
16:       s=s+"*";
17:     System.out.println(s);
18:   } // of method stars
19: } // of class methods2
```

```
1: import java.util.Scanner;
2: public class FindMin {
3:     public static void main( String args[] ){
4:         Scanner myIn = new Scanner(System.in);
5:         int num1, num2, num3, min;
6:         // read in the numbers from user
7:         System.out.println("Please enter 3 whole numbers ");
8:         num1 = myIn.nextInt();
9:         num2 = myIn.nextInt();
10:        num3 = myIn.nextInt();
11:        // call the minimum method
12:        min = mini( num1, num2, num3 );
13:        // display the result
14:        System.out.println("The minimum is " + min );
15:    }
16:    // minimum method definition
17:    public static int mini( int x, int y, int z ){
18:        int temp;
19:        if ( x < y)
20:            temp = x;
21:        else
22:            temp = y;
23:        if ( z < temp )
24:            return z;
25:        else
26:            return temp;
27:    }
28: }
```

```

1: //find all prime numbers in range given by user
2: import java.util.Scanner;
3: public class Primes {
4:     public static void main ( String arg[] ) {
5:         Scanner readStuff = new Scanner(System.in);
6:         int fromNum , toNum;
7:         System.out.println("Enter your range (2 numbers)");
8:         fromNum = readStuff.nextInt();
9:         toNum = readStuff.nextInt();
10:        String reply="Primes in range "+fromNum + " to "+toNum + " are:\n";
11:        for ( int i = fromNum; i <= toNum; i++ )
12:            if ( isPrime ( i ) )
13:                reply = reply+" " + i );
14:        // display output
15:        System.out.println(reply);
16:    }
17:    public static boolean isPrime ( int number ){
18:        double sqrt = Math.sqrt( number ); // compute the square root one time only!
19:        for ( int j=2; j <= sqrt; j++ ){
20:            if ( number % j == 0 )
21:                return false;
22:        }
23:        return true;
24:    }
25: }

```

תרגיל 8

כתבו מחלקה ובה השיטות הבאות:

1. `public static void twoRandNums(){...}`

השיטה מציגה על המסך שני מספרים אקראיים שלמים שונים בין 1 לבין 10.

2. `public static void twoRandNums(int toNum){...}`

השיטה מציגה על המסך שני מספרים אקראיים שלמים שונים בין 1 לבין toNum

3. `public static void twoRandNums(int fromNum ,int toNum){...}`

השיטה מציגה על המסך שני מספרים אקראיים שלמים שונים בין fromNum לבין toNum

4. `public static void fibo(int numbers){...}`

השיטה מציגה על המסך את numbers המספרים הראשונים בסדרת פיבונאצ'י. סדרת פיבונאצ'י היא סדרת מספרים שאיבריה הראשונים הם 1 ו-1, וכל איבר אחר בה שווה לסכום שני קודמיו. לדוגמה, 10 האיברים הראשונים של הסידרה הם: 1,1,2,3,5,8,13,21,34,55.

5. `public static void drawRectangle (int rows, int cols){...}`

השיטה מציירת מלבן המורכב מכוכביות ובו rows שורות ו-cols עמודות. לדוגמה אם מפעילים את השיטה כך:

```
drawRectangle(3,5)
```

מקבלים על המסך:

```
*****
*****
*****
```

ואם מפעילים את השיטה כך:

```
drawRectangle(4,4)
```

מקבלים על המסך:

```
****
****
****
****
```

6. `public static void drawTriangle (int rows){...}`

השיטה מציירת משולש המורכב מכוכביות ובו rows שורות. לדוגמה אם מפעילים את השיטה כך:

```
drawTriangle (5)
```

מקבלים על המסך:

```
*
**
***
****
*****
```

7. `public static void countLetterInSentence(String sentence, char let){...}`

השיטה מקבלת משפט ואת ומדפיסה את מספר הפעמים שהאות let מופיעה במשפט sentence.

כתבו את השיטה main והפעילו מתוכה את השיטות שכתבתם באופן הבא:

```
twoRandNums();  
twoRandNums(100);  
twoRandNums(20,40);  
fibonacci(15);  
drawRectangle(3,5)  
drawTriangle (5);
```

את השיטה 7 הפעילו כך:

```
countLetterInSentence("How are you today?","s");  
countLetterInSentence("How are you today?","a");  
countLetterInSentence("How are you today?","w");  
countLetterInSentence("How are you today?","x");
```

בתרגיל זה אין צורך לקרוא קלט מן המסך. הדפיסו את הפלט על המסך בעזרת System.out הגישו הדפסה של המחלקה שכתבתם ושל הפלט.

```

1:  import javax.swing.*;
2:  public class Cat{
3:      private String color;
4:      private String meow;
5:      private double weight;
6:      public Cat(){
7:          color="gray";
8:          meow="meowww";
9:          weight=1.0;
10:     }
11:     public Cat(String col,double w){
12:         color=col;
13:         meow="meoww";
14:         weight=w;
15:     }
16:     public void eat(double food){
17:         weight=weight+food/10;
18:     }
19:     public void lose (double loss){
20:         if (loss>0)
21:             weight=weight-loss;
22:     }
23:     public void speak(){
24:         JOptionPane.showMessageDialog(null,meow);
25:     }
26:     public double yourWeight (){
27:     return weight;
28:     }
29: }
-----
30:  import javax.swing.*;
31:  public class GoodCat{
32:      public static void main(String[] args){
33:          Cat mitzi = new Cat();
34:          Cat ziva = new Cat ("Black",2.5);
35:          mitzi.eat(0.5);
36:          double mishkal = mitzi.yourWeight();
37:          JOptionPane.showMessageDialog(null,"Cat's weight is "+mishkal);
38:          ziva.speak();
39:     }
40: }

```

משתני מחלקה

בנאי
אחד

בנאי שני


```
1: // Definition of class Die by Ori Keren
2: public class Die {
3:     private int value; // die current value
4:     private int max; // how many faces does the die have
5: // constructor
6: public Die(int m){
7:     max = m;
8:     roll(); // roll the die one time ..
9: }
10: public void roll(){
11:     value = (int)(max * Math.random()+1);
12: }
13: public int getVal(){
14:     return value;
15: }
16: public boolean is_equal(Die other_die){
17:     if (value == other_die.value)
18:         return true;
19:     else
20:         return false;
21: }
22: // clone current die;
23: public Die clone_die(){
24:     Die new_die = new Die(max);
25:     new_die.value = value;
26:     return new_die;
27: }
28: }
```

```

1: public class dice {
2:     public static void main( String args[] ){
3:         int sum;
6:         Die die1, die2; // declare 2 dice
7:         // initialize the dice
8:         die1 = new Die(6);
9:         die2 = new Die(6);
11:         die1.roll();
12:         die2.roll();
13:         if(die1.getVal()+die2.getVal(>9 )
14:             System.out.println("Sum of dice is greater than 9");
15:         else
16:             System.out.println("Sum of dice is less than or equal to 9");
18:         if (die1.is_equal(die2))
19:             System.out.println("Both dice show the same number");
20:         else
21:             System.out.println("Dice show different numbers");
22:     }
23: }

```

```

1: public class freq {
2:     public static void main(String[] args) {
3:         // creat a 3 sided die
4:         Die d1 = new Die(3);
5:         int mone1=0, mone2=0, mone3=0;
6:         // roll die 100 times, count how many times each number came up
7:         for(int i = 1 ; i <= 100 ; i++){
8:             d1.roll();
9:             switch (d1.getVal()){
10:                 case 1: mone1++;
11:                     break;
12:                 case 2: mone2++;
13:                     break;
14:                 case 3: mone3++;
15:             } // of switch
16:         } // of for
17:         System.out.println("number
freq\n1\t"+mone1+"\n2\t"+mone2+"\n3\t"+mone3);
18:     }
20: }

```

תרגיל 9

הכינו מחלקה Rectangle לייצוג מלבן. משתני המחלקה הם אורך המלבן ורוחב המלבן (מספרים שלמים) וערך יחיד מסוג char והוא הצורה באמצעותה ניתן לצייר את המלבן (הסבר בהמשך). שימו לב: למען הנוחות הוגדר צד אחד של המלבן בשם אורך וצד שני בשם רוחב למרות שיתכנו מקרים בהם ה"אורך" יהיה קצר מה"רוחב".

הכינו שני constructors:

1. ללא פרמטרים אשר יאתחל את האורך והרוחב לערך 1 ואת המשתנה השלישי לסימן *.

2. בעל 3 פרמטרים, אחד לכל משתנה.

הכינו את השיטות הציבוריות (public) הבאות:

public void drawRect() (1)

השיטה מדפיסה את המלבן בחלון בעזרת התו במשתנה השלישי. לדוגמה: עבור אורך 2 ורוחב 5 והסימן # תדפיס השיטה:

```
#####
```

```
#####
```

public int getLength() (2)

השיטה מחזירה את אורך המלבן

public void mult(int times) (3)

השיטה מקבלת מספר שלם כפרמטר ומשנה את אורכו ורוחבו של המלבן כך שיהיו times כפול האורך והרוחב המקורי.

public void biggest (Rectangle mal1, Rectangle mal2) (4)

השיטה מקבלת כפרמטרים שני מלבנים, ומשנה את מימדי המלבן עליו מופעלת השיטה כך שיהיו בעלי האורך הגדול מבין שני האורכים של המלבנים והרוחב יהיה הגדול מבין שני המלבנים. השיטה אינה משנה את התו (הצורה) של המלבן. דוגמה: אם מתקבל המלבן 7X5 ו 2X12 יש להציב אורך 7 ורוחב 12.

כתבו מחלקה בשם tar9 ובה בדקו המחלקה הנ"ל באופן הבא:

- הגדירו את 3 האובייקטים הבאים מסוג מלבן:

צורה	רוחב	אורך
X	5	7
\$	12	2
@	2	3

- בעזרת השיטה שכתבתם, הדפיסו את מלבן הראשון.
 - בעזרת השיטה השלישית, הגדילו את מימדי המלבן השלישי עם פרמטר שהוא אורך המלבן השני. הדפיסו את המלבן המתקבל.
 - הפעילו על המלבן השלישי את השיטה biggest כשהפרמטרים הם המלבן הראשון והמלבן השני. והדפיסו את המלבן המתקבל.
- הדפיסו והגישו את שתי המחלקות ואת הפלט. שדכו דפים.

מערכים

משתנה מסוג מערך ניתן להציג באופן הבא:

0	1	2	3	4	5	6	7	8	9

למערך שם אחד ולכל תא במערך יש מצין (אינדקס).

הגדרת המערך:

; שם המערך [] סוג אברי המערך

דוגמה להגדרת המערך הנ"ל:

```
int [] numbers;
```

ואח"כ הקצאת מקום בזיכרון למערך:

```
[מספר תאי המערך] סוג תאי המערך = new
```

דוגמה להקצאת מקום בזיכרון:

```
numbers = new int [10];
```

הוגדר מערך בשם numbers בעל 10 תאים הממוספרים מ 0 עד 9. התאים יכולו ערכים מספריים שלמים.

הצבת ערכים בתאי המערך:

```
numbers[3]=15;
```

```
numbers[0]=3;
```

```
numbers[9]=-4;
```

לאחר ביצוע ההוראות הנ"ל יראה המערך כך:

0	1	2	3	4	5	6	7	8	9
3	0	0	15	0	0	0	0	0	-4

בדרך כלל יהיה האינדקס משתנה ולא קבוע:

```
int i = 2;
```

```
numbers[i]=10;
```

```
numbers[i+1]=19;
```

לאחר ביצוע ההוראות הנ"ל יראה המערך כך:

0	1	2	3	4	5	6	7	8	9
3	0	10	19	0	0	0	0	0	-4

דוגמה נוספת:

```
for (i=0; i<10;i++) {
```

```
numbers[i]=i+3;
```

```
}
```

לאחר ביצוע ההוראות הנ"ל יראה המערך כך:

0	1	2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10	11	12

!! אסור לחרוג מהתחום שהוגדר עבור המערך !!

מותר להגדיר מערך ולהקצות לו זיכרון ולאתחל אותו באופן הבא:

```
int numbers [] = {5, 8, 8, 4, 9, 67, 100, 0, -6, 5};
```

הוגדר מערך של מספרים שלמים בגודל 10 ובו הערכים הנ"ל.

```

1:  import java.util.Scanner;
2:  public class Arrays1{
3:  public static void main(String[] args){
4:  Scanner read = new Scanner(System.in);
5:  final int STUDENTS=5;
6:  int sum=0,i,count=0;
7:  double average;
8:  String s;
9:  int [] marks;
10: marks=new int[STUDENTS];
11: System.out.println("Please enter "+STUDENTS+" grades");
12: for (i=0;i<STUDENTS;i++){
13:     marks[i] = read.nextInt();
14:     sum=sum+marks[i];
15: }
16: average=(double)sum/STUDENTS;
17: s="Average is: "+average+"\nAbove average grades:\n";
18: for (i=0;i<STUDENTS;i++)
19:     if(marks[i]>average){
20:         count++;
21:         s=s+marks[i]+" ";
22:     }
23: s=s+"\n "+count+" grades";
24: System.out.println(s);
25: }
26: }

```

----- Class TestArrays2 -----

```

0:  import java.util.Scanner;
1:  public class TestArrays2 {
2:  public static void main(String[] args){
3:  Arrays2 ar = new Arrays2(20);
4:  ar.displayNums();
5:  int sum = ar.sumNums();
6:  int max = ar.findMax();
7:  int cell = ar.findPlaceOfMax();
8:  System.out.println("Sum="+sum+" Max="+max+" in cell "+cell);
9:  }
10: }

```

```

1: public class Arrays2{
3:     private int [] maar;
5:     public Arrays2 (int size){ // constructor
6:         maar=new int[size + 1];
7:         // enter numbers into array from cell 1 to cell size
8:         for (int i=1;i<=size;i++)
9:             maar[i]=(int) (Math.random()*100) ;
10:    }
12:
13:    public void displayNums() { //Display numbers
14:        String s=(" Numbers are:\n");
15:        for (int i=1;i<maar.length;i++)
16:            if (i% 4 ==0)
17:                s=s+maar[i]+" \n";
18:            else
19:                s=s+maar[i]+" \t";
20:        System.out.println(s);
21:    }
23:    public int sumNums (){ // find sum of numbers in array
24:        int sum =0;
25:        for (int i=1;i<maar.length;i++)
26:            sum=sum+maar[i];
27:        return sum;
28:    }
30:    public int findMax(){ //find maximum of numbers in array
31:        int max = -1;
32:        for (int i=1;i<maar.length;i++)
33:            if (maar[i]>max)
34:                max=maar[i];
35:        return max;
36:    }
38:    public int findPlaceOfMax(){ //find place of maximum of numbers in array
39:        int place=1;
40:        int max=maar[1];    // max variable is not necessary
41:        for (int i=2;i<maar.length;i++)
42:            if (maar[i]>max){
43:                max=maar[i];
44:                place=i;
45:            }
46:        return place;
47:    }
48: }

```

```
1: public class FreqArray {
2:     public static void main(String[] args) {
3:         // creat a 6 sided die
4:         Die d1 = new Die(6);
5:         int [] mone = new int [7];
6:         // roll die 100 times, count how many times each number came up
7:         for(int i = 1 ; i <= 100 ; i++){
8:             d1.roll();
9:             int face =d1.getVal();
10:            mone[face]=mone[face]+1;
11:        } // of for
12:        System.out.println("Number freq");
13:        for (int i = 1; i <= 6; i++)
14:            System.out.println(i+"\t"+mone[i]);
15:    }
16: }
```

```

1: // A palyndrom
2:
3:  import javax.swing.*;
4: public class TestPalyndrom {
5     public static void main(String[] args) {
6         String s = JOptionPane.showInputDialog("Your word please");
7         Palyndrom p = new Palyndrom(s);
8         boolean reply = p.check();
9         if (reply == true)
10:             JOptionPane.showMessageDialog(null ,"It is a Palyndrom",100,100);
11:         else
12:             JOptionPane.showMessageDialog(null ,"It is not a Palyndrom",100,100);
13:     }
14: }

```

```

18: public class Palyndrom {
19:     private String word;
20:     public Palyndrom(String s) { // constructor
21:         word = s;
22:     }
23:     public boolean check(){ // check if word is a palyndrom
24:         char [] letters;
25:         letters = word.toCharArray();
26:         for (int i = 0, j =word.length()-1; i<j ; i++,j--)
27:             if (letters[i]!=letters[j])
28:                 return false;
29:         return true;
30:     }
31: }

```


תרגיל 10

הכינו מחלקה MyArray. משתנה המחלקה הוא מערך של מספרים שלמים בשם nums :

```
private int [] nums;
```

הכינו בנאי (constructor) בעל פרמטר אחד אשר יקבע את גודל המערך. הבנאי יצור את המערך בגודל המבוקש ויציב בו מספרים אקראיים שלמים בטווח 0 עד 100.

הכינו את השיטות הציבוריות (public) הבאות:

```
1. public void displayArray()
```

השיטה מציגה את אברי המערך בחלון.

```
2. public int countUnder (int number)
```

השיטה מחזירה את מספר המספרים במערך הקטנים מ "number" (הפרמטר שהועבר).

```
3. public boolean sidraOla (int upTo)
```

השיטה מחזירה TRUE אם המספרים במערך עד לתא upTo מהוים סדרת מספרים עולה (כל מספר גדול מקודמו), אחרת היא מחזירה FALSE.

```
4. public boolean findNumber (int number)
```

השיטה מחפשת את המספר number (שהועבר כפרמטר) במערך ומחזירה TRUE אם הוא נמצא במערך, FALSE אחרת.

כתבו מחלקה בשם Tar10 ובה השיטה main הבודקת את המחלקה באופן הבא:

מגדירה אובייקט מסוג MyArray.

מפעילה כל אחת מ 4 השיטות הנ"ל. עבור שיטה 2 יש להעביר פרמטר מספרי (למשל 50), עבור השיטה 3 יש להעביר פרמטר (למשל 2) ועבור שיטה 4 יש להעביר פרמטר (למשל 37). לאחר הפעלת שיטות 2-4 יש להציג בחלון את התוצאות שהתקבלו.

הדפיסו והגישו את שתי המחלקות ואת הפלט. שדכו דפים.

מערכים דו-ממדיים

הגדרה:

```
int mat [] [];
```

הקצאת זיכרון:

```
mat = new int [3] [5];
```

או הגדרה והקצאת זיכרון ואיתחול:

```
int mat [] [] = {{1,5,3,7,8}, {6,6,6,6,6}, {78,34,56,2,0}};
```

ההגדרה הנ"ל (ללא אתחול) יוצרת את המערך הדו-ממדי הבא בעל 3 שורות ו 5 עמודות (המספר הראשון קובע מספר שורות והמספר השני קובע מספר עמודות):

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0

הצבת ערך בתא: `mat [1] [4] = 11;`

הוראה זו מציבה 11 בתא הנמצא בשורה מספר 1 (השורה השנייה) ובעמודה מספר ארבע (עמודה חמישית). לאחר ההצבה יראה המערך כך:

0	0	0	0	0
0	0	0	0	11
0	0	0	0	0

התייחסות לתא מסוים נעשית באמצעות שם המערך, אחריו מספר השורה ואחריו מספר העמודה בסוגריים מרובעים.

כדי לעבור על כל תאי מערך דו-ממדי, נוח להשתמש בלולאות כפולות. בדוגמה הבאה עוברים על אברי המערך הנ"ל ובודקים אם יש בו אברים שליליים:

```
for (int i=0; i<3; i++)
  for (int j=0; j<5; j++)
    if (mat[i][j] < 0)
      System.out.println( " There is a negative number " );
```

```
1: // Students' grades in tests.
2:  import java.util.Scanner;
3:  public class UseMatrix {
4:    public static void main(String[] args) {
5:      Scanner s = new Scanner(System.in);
6:      // Use class Matrix to hold student's grades in tests.
7:      System.out.println("Please enter number of students and number of tests");
8:      int students = s.nextInt();
9:      int tests = s.nextInt();
10:     Matrix grades = new Matrix(students, tests);
11:     // Read in student's grades
```

```

12:     for (int i = 0; i < students; i++){
13:         System.out.println("please enter grades for student "+i);
14:         for (int j = 0 ; j < tests; j++){
15:             int g = s.nextInt();
16:             grades.enterNum(i, j, g);
17:         }
18:     }           //Display a student's average
20:     System.out.println("Whose average do you need?");
21:     int stud = s.nextInt();
22:     int sumForStud=grades.sumOfRow(stud);
23:     double avgStud = (double)(sumForStud)/tests;
24:     System.out.println("Average of grades for student "+stud+" is "+avgStud);
25:     // Display the average for a certain test
26:     System.out.println("Average of which test do you need ");
27:     int test = s.nextInt();
28:     int sumForTest=grades.sumOfCol(test);
29:     double avgTest = (double)(sumForTest)/students;;
30:     System.out.println("Average for test "+test+" is "+avgTest);
31: }
32: }

```

```

1: // Using a 2 dimensional array
2: public class Matrix {
3:     private int [][] mat;
4:     public Matrix(int row, int col){
5:         mat=new int[row][col];
6:     }
7:     public void enterNum (int r, int c, int num){
8:         mat[r][c] = num;
9:     }
10:    public int sumOfCol (int col){    // Compute sum of Column
11:        int sum=0;
12:        for (int r=0;r<mat.length;r++)
13:            sum=sum+mat[r][col];
14:        return sum;
15:    }
16:    public int sumOfRow (int row){    // Compute sum of row
17:        int sum=0;
18:        for (int c=0;c<mat[0].length;c++)
19:            sum=sum+mat[row][c];
20:        return sum;
21:    }
22: }

```

```

// Build a magic square
public class Magic {
    private int [][]mag;
1:   public Magic(int size) { // size must be an odd number
2:       mag = new int [size][size];
3:       // Create magic square
4:       int i=0;
5:       int j=size/2;
6:       mag[i][j]=1;
7:       for (int x=2;x<=size*size;x++) {
8:           int otekI=i;
9:           int otekJ=j;
10:          i=i-1; //move up one line
11:          if (i<0) // if out of bounds
12:              i=size-1;
13:          j=j+1; //move right one column
14:          if (j>=size) // if out of bounds
15:              j=0;
16:          if (mag[i][j] != 0){ // if place is taken
17:              i=otekI; // go back to former position
18:              j=otekJ;
19:              i++; // move one line down
20:          } // end of if
21:          mag[i][j]=x; // found new place, insert number
22:      } // end of for loop
23:  }
24:  public void displayMagic(){
25:      for (int i=0;i<mag.length;i++) {
26:          for (int j=0;j<mag[0].length;j++)
27:              System.out.print(mag[i][j]+"\\t");
28:          System.out.println();
29:      } // of for loop
30:  }
31: }

```

```

1: // Use class Magic
3: public class PlayMagic {
4:     public static void main(String[] args) {
5:         Magic m = new Magic(7);
6:         m.displayMagic();
7:     }
8: }

```

כתבו מחלקה בשם `TwoDimArray`. משתנה המחלקה יהיה מערך דו-ממדי בו מספר השורות שווה למספר העמודות והינו מספר אי-זוגי.

הבנאי של המחלקה יקבל כפרמטר את מספר השורות/העמודות במערך ויבדוק אם המספר אי-זוגי. אם אינו אי-זוגי, יוסיף לו הבנאי 1 וכך יהפוך אותו לאי-זוגי. הבנאי יצור את המערך וימלא את התאים במספרים אקראיים בין 0 ל 100.

המחלקה תגדיר את 3 השיטות הבאות:

השיטה `public void print()` תציג בחלון הפלט את המספרים המאוכסנים במערך.

השיטה `public void maxOfRows()` תציג בחלון הפלט, עבור כל שורה במערך, את המספר הגבוה ביותר באותה שורה ובאיזה עמודה נמצא מספר זה (אם המספר מופיע יותר מפעם אחת, השיטה תבחר עמודה כלשהי בה מופיע המספר).

השיטה `public int sumOfDiagonal()` תחזיר את סכום המספרים הנמצאים על האלכסון הראשי של המערך.

לדוגמה עבור מערך של 3X3. אם המערך מכיל את הנתונים הבאים:

4	2	5
17	20	11
8	7	8

תציג השיטה `maxOfRows` את התוצאות הבאות:

```
Line 0 max=5 in column 2
Line 1 max=20 in column 1
Line 2 max=8 in column 0
```

והשיטה `sumOfDiagonal` תחזיר את המספר 32 כי $4+20+8=32$.

כתבו מחלקה בשם `Tar11` ובה שיטה `main` היוצרת אובייקט של המחלקה `TwoDimArray` עם הפרמטר 5 ומפעילה על האובייקט את 3 השיטות של המחלקה.

הגישו הדפסה של שתי המחלקות ושל חלון הפלט. שדכו דפים.

```

1: // Sorting and searching an array
2: public class Arrayservice {
3:     private int [] a;    // the array's size is unknown
4: // constructor
5: // The size is determined by the client
6: public Arrayservice(int size) {
7:     a = new int [size];
8: }
9: // put element into place in the array
10: public void putElement( int element, int place )
11: {
12:     a [place] = element;
13: }
14: //print the array
15: public void print()
16: {
17:     for ( int i=0; i<a.length; i++ )
18:         System.out.print (" " + a[i] );
19:     System.out.println();
20: }
21: // find number in unsorted array, if not found return -1
22: public int searchUnsorted (int element) {
23:     for (int i = 0; i < a.length; i++)
24:         if (a[i] == element)
25:             return i;
26:     return -1;
27: }
28: //sort the array
29: public void sort() { // uses selection sort
30:     for ( int i=0; i<a.length; i++)
31:     {
32:         // find the position of the smallest value
33:         // of the sublist a[i]... a[a.length - 1]
34:         int min = i;
35:         for (int j=i+1; j<a.length; j++ )
36:             if (a[j] < a[min])
37:                 min = j;
38:         // put the smallest value at the beginning
39:         // of the sublist
40:         swap (i,min);
41:     } // of for loop
42: }
43: private void swap ( int i, int j) // is a private method
44: {
45:     int temp = a[i];

```

```

46:  a[i] = a[j];
47:  a[j] = temp;
48: }
49: // find number in sorted array, if not found return -1
50: public int searchSorted (int element) {
51:     for (int i = 0; i < a.length; i++)
52:         if (a[i] == element)
53:             return i;
54:         else if (a[i] > element)
55:             return -1;
56:     return -1;    // element greater than last number in array
57: }
58: // find number in sorted array using binary search, if not found return -1
59: public int binarySearch (int element) {
60:     int start = 0;
61:     int end = a.length - 1;
62:     while (start <= end) {
63:         int mid = (start + end) / 2;
64:         if (element == a[mid])
65:             return mid;
66:         else if (element < a[mid])
67:             end = mid - 1;
68:         else
69:             start = mid + 1;
70:     }
71:     return -1;
72: }
73: }

```

```

1: // search for a convict
2: import java.util.Scanner;
3: public class Convicts {
4:     public static void main(String[] args) {
5:         Scanner input = new Scanner(System.in);
6:         final int SIZE = 4;
7:         Arrayservice ids = new Arrayservice(SIZE);
8:         // read in ids of convicts and put in array
9:         for (int i = 0; i < SIZE ; i++) {
10:             System.out.println("Id please");
11:             int id = input.nextInt();
12:             ids.putElement(id,i);
13:         }
14:         ids.print();
15:         // Read id of person and check if he is a convict
16:         System.out.println(" New Id please");
17:         int suspect = input.nextInt();
18:
19:         // Check if suspect was a convict
20:         int answer = ids.searchUnsorted(suspect);
21:         if (answer == -1)
22:             System.out.println("First arrest");
23:         else
24:             System.out.println("Was arrested in the past");
25:         // Sort the array
26:         ids.sort();
27:         ids.print();
28:         // Check if suspect was a convict using sorted array
29:         answer = ids.searchSorted(suspect);
30:         if (answer == -1)
31:             System.out.println("First arrest");
32:         else
33:             System.out.println("Was arrested in the past");
34:         // Check if suspect was a convict using sorted array - binary search
35:         answer = ids.binarySearch(suspect);
36:         if (answer == -1)
37:             System.out.println("First arrest");
38:         else
39:             System.out.println("Was arrested in the past");
40:     }
41: }

```



```
1: import java.io.*;
2: import java.util.Scanner;
3: public class ReadFromFile // read name from a file
4: {
5: public static void main(String[] args) throws IOException
6: {
7: File myFile = new File("data.txt");
8: Scanner myF = new Scanner(myFile);
9: String s;
10: s=myF.nextLine();
11: System.out.println("Hello " + s);
12: }
13: }
```

```
1: import java.io.*;
2: import java.util.Scanner;
3: public class ReadFromFile1 // read names from a file till end of file
4: {
5: public static void main(String[] args) throws IOException
6: {
7: File myFile = new File("data.txt");
8: Scanner myF = new Scanner(myFile);
9: String s;
10: while (myF.hasNext()){
11: s=myF.nextLine();
12: System.out.println("Hello " + s);
13: }
14: }
15: }
```

```

1: import java.io.*;
2: import java.util.Scanner;
3: public class WriteToFile // Read numbers, write their square root to a file
4: {
5: public static void main(String[] args) throws IOException
6: {
7: File inFile = new File("inData.txt");
8: Scanner inF = new Scanner(inFile);
9:
10: FileWriter outFile = new FileWriter("outData.txt");
11: PrintWriter outF = new PrintWriter(outFile);
12:
13: double num,numX;
14: while (inF.hasNext()){
15:     num=inF.nextDouble();
16:     numX=Math.sqrt(num);
17:     outF.println("Sqrt of "+ num + " is "+ numX);;
18: }
19: outF.close();
20: }
21: }

```

```

1: // Calculates student's grades
2: import java.io.*;
3: import java.util.Scanner;
4: public class StudentGrades {
5: private int [] courses;
6: private int [] credits;
7: private int [] grades;
8: private int size;
9:// constructor receives the name of the file which holds the information about the student
10: public StudentGrades(String nameOfFile) throws IOException{
11: File myFile = new File(nameOfFile);
12: Scanner myF = new Scanner(myFile);
13: //create 100 cell arrays assuming no more than 100 courses were taken by student
14: courses = new int [100];
15: credits=new int [100] ;
16: grades = new int [100] ;
17: int i =0;
18: while (myF.hasNext()){

```

```

19:     courses[i]=myF.nextInt();
20:     credits[i]=myF.nextInt();
21:     grades[i]=myF.nextInt();
22:     i++;
23: }
24: size = i;
25: }
26: public double average(){ // Returns student's average grade
27:     double sumCredits = 0, sum=0;
28:     for (int i = 0; i<size ; i++){
29:         sumCredits += credits[i];
30:         sum +=credits[i] * grades[i];
31:     }
32:     return sum / sumCredits;
33: }
34: public int sumOfCredits(){ //Returns sum of credits student took
35:     int sumCredits = 0;
36:     for (int i = 0; i<size ; i++)
37:         sumCredits += credits[i];
38:     return sumCredits;
39: }
40: //Calculate average of grades of courses taken in a specific deptment
41: public double specificAverage(int dep){
42:     double sumCredits = 0, sum=0;
43:     for (int i = 0; i<size ; i++)
44:         if (courses[i]/100 == dep){
45:             sumCredits += credits[i];
46:             sum +=credits[i] * grades[i];
47:         }
48:     if (sumCredits!=0)
49:         return sum / sumCredits;
50:     else
51:         return -1; //if no courses were taken for the specific department, return -1
52: }
53: //returns sum of credits taken in a specific deptment
54: public int specificSumOfCredits (int dep){
55:     int sumCredits = 0;
56:     for (int i = 0; i<size ; i++)
57:         if (courses[i]/100 == dep)
58:             sumCredits += credits[i];
59:     return sumCredits;
60:
61: }
62: }

```

```

1: // Use class StudentGrades
2: import java.io.*;
3: import java.util.Scanner;
4: public class MyGrades {
5:     public static void main(String[] args) throws IOException {
6:         Scanner readIn = new Scanner(System.in);
7:         System.out.println("Hello dear student, what is the name of your file");
8:         System.out.println("and what would you like from us?");
9:         System.out.println("1 = to see the sum of your credits");
10:        System.out.println("2 = to see your average");
11:        System.out.println
12:            ("3 = to see the sum of your credits in a certain department (enter dep code)");
13:        System.out.println
14:            ("4 = to see your average in a certain department (enter dep code)");
15:        String name = readIn.next();
16:        int choice = readIn.nextInt();
17:        StudentGrades myStud = new StudentGrades(name);
18:        double avg;
19:        int sum, dep;
20:        switch (choice){
21:            case 1: sum = myStud.sumOfCredits();
22:                System.out.println("The sum is "+sum );
23:                break;
24:            case 2: avg = myStud.average();
25:                System.out.println("The average is "+avg );
26:                break;
27:            case 3: dep = readIn.nextInt();
28:                sum = myStud.specificSumOfCredits(dep) ;
29:                System.out.println("The sum is "+sum );
30:                break;
31:            case 4: dep = readIn.nextInt();
32:                avg = myStud.specificAverage(dep);
33:                System.out.println("The average is "+avg );
34:                break;
35:        }
36: }

```

תרגיל 12

כתבו את הפתרון לתרגל זה בכתב ולא כתכנית מחשב.

במחלקה StudentGrades מוגדרים שני משתנים:

1. מערך דו-ממדי של מספרים שלמים בשם grades : `int [][] grades`
2. מערך חד-ממדי של תווים בשם gender : `char [] gender`

כל שורה במערך grades מיצגת תלמיד אחד, וכל עמודה מיצגת מקצוע לימוד אחד. תאי המערך מכילים ציונים כך שציונו של התלמיד ה- i במקצוע j מופיע בתא הנמצא השורה i בעמודה j במערך. לדוגמה, ציונו של התלמיד הראשון במקצוע הראשון מופיע במערך בתא [0][0], ציונו במקצוע השני מופיע בתא [0][1] (שורה אפס, עמודה אחת) וכך הלאה. המערך gender מכיל את מין התלמידים. מינו של התלמיד ה-i מופיע בתא ה-i. לדוגמה, מינו של התלמיד הראשון מופיע בתא [0], מינו של התלמיד השני מופיע בתא [1] וכך הלאה.

הניחו שהבנאי כבר כתוב והוא הציב לכל תלמיד את ציוניו במערך grades ואת מינו במערך gender

עליכם לכתוב שתי שיטות נוספות עבור המחלקה StudentGrades:

השיטה

public double average (int grade, char sex)

הפרמטר grade יכול מספר שלם בין 0 ל-100. הפרמטר sex יכול ערך 'M' או 'F'. השיטה תחשב ותחזיר את אחוז התלמידים שהמין שלהם הוא sex מבין התלמידים שהציון הממוצע שלהם הוא לפחות grade. אם אין תלמידים שהציון הממוצע שלהם הוא לפחות grade, השיטה תחזיר -1.

השיטה

public double percentSex (char sex)

הפרמטר sex יכול ערך 'M' או 'F'. השיטה תחשב ותחזיר את אחוז התלמידים שקיבלו לפחות פעם אחת 100 מבין התלמידים שהמין שלהם הוא sex. ניתן להניח שתמיד יהיו תלמידים שהמין שלהם הוא sex.

כתבו את ההוראות הבאות עבור השיטה main:

צרו אובייקט מסוג StudentGrades,

הפעילו על האובייקט את שתי השיטות שכתבתם עם פרמטרים כרצונכם והציגו את התוצאות.

תרגיל 13

כתבו תוכנית ובה 2 מחלקות. המחלקה **Targil13** והמחלקה **TestTargil13** אשר מכילה את השיטה `main` ומשתמשת במחלקה **Targil13**.

להלן השורות הראשונות של המחלקה **Targil13** כולל הגדרת המשתנים של המחלקה:

```
Public class Targil13 {
private int[ ][ ] table;
private int[ ] numbers;
```

המערך `table` הוא מערך דו מימדי ריבועי בו 25 שורות ו 25 עמודות.
כל תא במערך `table` מכיל מספר שלם בין 1 ל 100 .
המערך `numbers` הוא מערך חד מימדי של מספרים שלמים. במערך 60 תאים.

בצעו את הפעולות הבאות :

1. הכינו את המחלקה **Targil13** ובה שלוש השורות המופיעות מעלה ואחריהן:

```
public Targil13 ()
```

1) כתבו בנאי עבור המחלקה **Targil13** שכתרתו `public Targil13 ()` ובו הקצו זכרון לשני המערכים `table` ו-`numbers`, הציבו מספרים שלמים בין 1 לבין 100 במערך `table` והפעילו את השיטה `fillRandNumbers` המתוארת בסעיף 2.

2) כתבו את השיטה

```
public void fillRandNumbers ( ) {...}
```

במערך `numbers` 60 תאים.
השיטה תציב ערכים במערך באופן הבא:
בתא הראשון במערך יש להציב אפס.
אח"כ תעבור השיטה על שאר תאי המערך `numbers` ותציב בתא ה `i` מספר שלם אקראי בטווח `i .. i-1` .

3) כתבו את השיטה

```
public int minPositive( ) {...}
```

השיטה תימצא ותחזיר את הערך המינימלי מבין הערכים החיוביים במערך `numbers` .
4) כתבו את השיטה

```
public boolean rowIsBigger( int row, int col) {...}
```

השיטה מקבלת שני אינדקסים המציינים מקום של תא במערך `table` . מספר שורה `row` ומספר עמודה `col` .
השיטה תחזיר `true` אם תא זה מקיים שסכום השורה `row` בה הוא נמצא, גדול מסכום העמודה `col` במערך `table` . אחרת תחזיר השיטה `false` .

5 (כתבו את השיטה

```
public int aboveDiagonal( ){...}
```

השיטה סופרת ומחזירה את מספר התאים מעל האלכסון הראשי במערך table , המקיימים את התנאי שבשאלה 4 (שסכום השורה בה נמצא התא גדול מסכום העמודה בה נמצא התא).

```
public boolean rowsBigger( int row, int col)
```

הנחיה: עליכם להשתמש בשיטה שכתבתם בשאלה 4 .

השיטה תחזיר את מספר התאים שמקיימים את התנאי.

- ii. כתבו את המחלקה **TestTargil13** ובה השיטה main אשר תכיל את החלקים הבאים:
- הוראה היוצרת אובייקט של המחלקה **Targil13**. זכרו שהבנאי של המחלקה אינו זקוק לפרמטרים.
 - הגדרה של משתנים לאחסון הפרמטרים הדרושים להפעלת השיטות השונות של המחלקה **Targil13**. הצבת ערכים מתאימים במשתנים אלו. שימו לב, אין צורך לקרוא אותם מהמשתמש.
 - הגדרת משתנים לצורך אחסון התוצאות המוחזרות מהשיטות של שאלות 3 + 5 .
 - הוראות המפעילות את השיטות 3 + 5 ומציבות את התוצאות המוחזרות במשתנים שהגדרתם בסעיף ג.