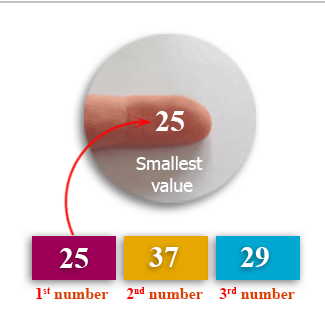
1.Write a Java method to find the smallest number among three numbers..

Test Data:  
Input the first number: 25  
Input the Second number: 37  
Input the third number: 29



import java.util.Scanner;

public class Exercise1 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input the first number: ");

double x = in.nextDouble();

System.out.print("Input the Second number: ");

double y = in.nextDouble();

System.out.print("Input the third number: ");

double z = in.nextDouble();

System.out.print("The smallest value is " + smallest(x, y, z)+"\n" );

}

public static double smallest(double x, double y, double z)

{

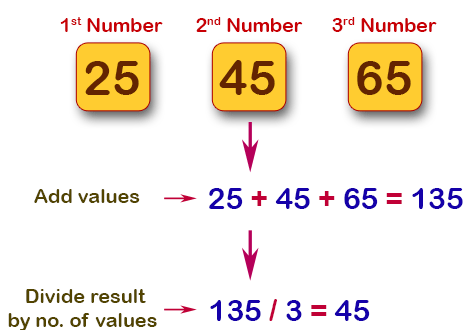
return Math.min(Math.min(x, y), z);

}

}

2.Write a Java method to compute the average of three numbers.

Test Data:  
Input the first number: 25  
Input the second number: 45  
Input the third number: 65



import java.util.Scanner;

public class Exercise2 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input the first number: ");

double x = in.nextDouble();

System.out.print("Input the second number: ");

double y = in.nextDouble();

System.out.print("Input the third number: ");

double z = in.nextDouble();

System.out.print("The average value is " + average(x, y, z)+"\n" );

}

public static double average(double x, double y, double z)

{

return (x + y + z) / 3;

}

}

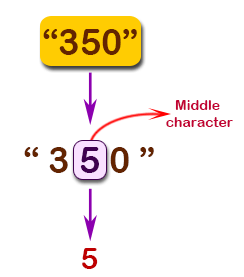
3.Write a Java method to display the middle character of a string.

**Note:** a) If the length of the string is odd there will be two middle characters.

b) If the length of the string is even there will be one middle character.

Test Data:

Input a string: 350



import java.util.Scanner;

public class Exercise3 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input a string: ");

String str = in.nextLine();

System.out.print("The middle character in the string: " + middle(str)+"\n");

}

public static String middle(String str)

{

int position;

int length;

if (str.length() % 2 == 0)

{

position = str.length() / 2 - 1;

length = 2;

}

else

{

position = str.length() / 2;

length = 1;

}

return str.substring(position, position + length);

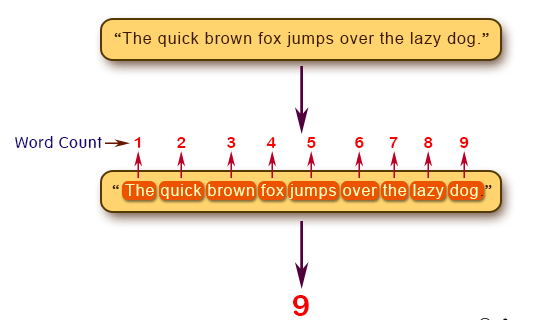
}

}

4.Write a Java method to count all words in a string.

Test Data:

Input the string: The quick brown fox jumps over the lazy dog.



import java.util.Scanner;

public class Exercise5 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input the string: ");

String str = in.nextLine();

System.out.print("Number of words in the string: " + count\_Words(str)+"\n");

}

public static int count\_Words(String str)

{

int count = 0;

if (!(" ".equals(str.substring(0, 1))) || !(" ".equals(str.substring(str.length() - 1))))

{

for (int i = 0; i < str.length(); i++)

{

if (str.charAt(i) == ' ')

{

count++;

}

}

count = count + 1;

}

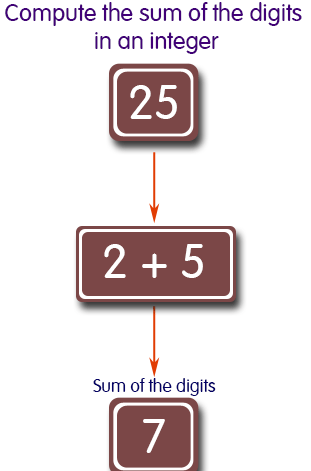
return count; // returns 0 if string starts or ends with space " ".

}

}

5.Write a Java method to compute the sum of the digits in an integer.

Test Data:  
Input an integer: 25



import java.util.Scanner;

public class Exercise6 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input an integer: ");

int digits = in.nextInt();

System.out.println("The sum is " + sumDigits(digits));

}

public static int sumDigits(long n) {

int result = 0;

while(n > 0) {

result += n % 10;

n /= 10;

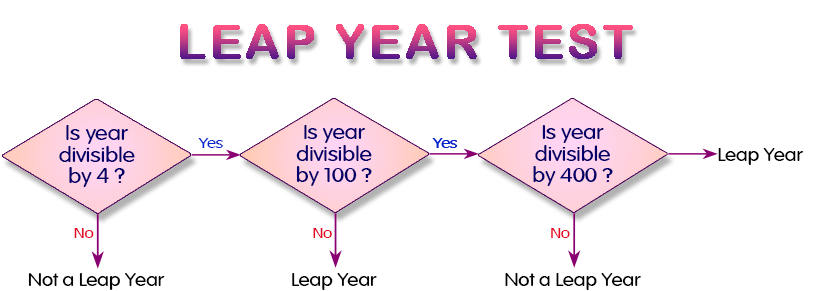
}

return result;

}

}

6.Write a Java method to check whether an year (integer) entered by the user is a leap year or not.



import java.util.Scanner;

public class Exercise10 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input a year: ");

int year = in.nextInt();

System.out.println(is\_LeapYear(year));

}

public static boolean is\_LeapYear(int y)

{

boolean a = (y % 4) == 0;

boolean b = (y % 100) != 0;

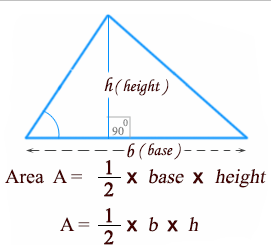
boolean c = ((y % 100 == 0) && (y % 400 == 0));

return a && (b || c);

}

}

7.Write Java methods to calculate the area of a triangle.



import java.util.Scanner;

public class Exercise13 {

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

System.out.print("Input Side-1: ");

double side1 = in.nextDouble();

System.out.print("Input Side-2: ");

double side2 = in.nextDouble();

System.out.print("Input Side-3: ");

double side3 = in.nextDouble();

System.out.println( is\_Valid(side1, side2,side3) ?

"The area of the triangle is " + area\_triangle(side1, side2, side3) : "Invalid triangle" );

}

public static boolean is\_Valid(double side1, double side2, double side3) {

if( side1 + side2 > side3 &&

side2 + side3 > side1 &&

side1 + side3 > side2) return true;

else return false;

}

public static double area\_triangle(double side1, double side2, double side3) {

double area = 0;

double s = (side1 + side2 + side3)/2;

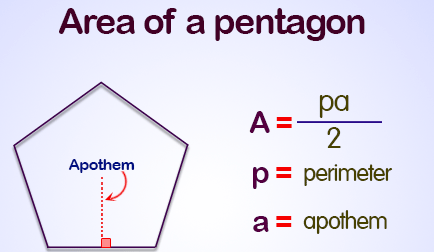
area = Math.sqrt(s\*(s - side1)\*(s - side2)\*(s - side3));

return area;

}

}

8.Write a Java method to create the area of a pentagon.



import java.util.Scanner;

public class Exercise14 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Input the number of sides: ");

int n = input.nextInt();

System.out.print("Input the side: ");

double side = input.nextDouble();

System.out.println("The area of the pentagon is " + pentagon\_area(n, side));

}

public static double pentagon\_area(int n, double s) {

return (n \* s \* s) / (4 \* Math.tan(Math.PI/n));

}

}