1. Faça uma função em Java que calcule e retorne o IMC de uma pessoa. Defina quais parâmetros devem ser passados para sua função.

2. Faça um procedimento em Java que receba por parâmetro um vetor e a opção de impressão (positiva imprime normalmente o vetor negativo imprime em ordem invertida)

import java.util.Scanner;  
import java.util.Random;  
public class Quest2{  
 public static void main(String[] args){  
 Scanner leia = new Scanner(System.in);  
 System.out.println("Digite numero positivo para ordem crescente e negativo para ordem decrescente");  
 byte op = leia.nextByte();  
 int VET[] = new int[5];  
 Vet(op,VET);  
   
 }  
 public static void Vet(byte op, int VET[])  
 {  
 Random r = new Random();  
 for(int i = 0; i< VET.length; i++)  
 {  
 VET[i] = r.nextInt(10);   
 }  
 if(op >= 0)  
 {  
 for(int i = 0; i < VET.length; i++)  
 {  
 System.out.print(VET[i] + " ");   
 }  
  
 }  
 else  
 for(int i = VET.length - 1; i >=0; i--)  
 {  
 System.out.print(VET[i] + " ");   
 }  
   
 }  
}

3. Faça um método que receba um vetor inteiro e o ordene.

import java.util.Random;  
public class Quest3{  
 public static void main(String[] args){  
 int VET[] = new int[5];  
 Vet(VET);   
 }  
   
 public static void Vet(int VET[])  
 {  
 Random r = new Random();  
 int aux;  
 for(int i = 0 ; i< VET.length ; i++)  
 {  
 VET[i] = r.nextInt(10);  
 }  
   
 for(int i = 0 ; i< VET.length ; i++)  
 {  
 for(int j = 0 ; j< VET.length ; j++)  
 {  
   
 if(VET[i] < VET[j])  
 {  
 aux = VET[i];  
 VET[i] = VET[j];  
 VET[j] = aux;  
 }  
 }  
 }  
   
 for(int i = 0 ; i< VET.length ; i++)  
 {  
 System.out.print(VET[i] + " ");  
 }   
   
 }  
}

4. Faça uma função que receba uma matriz e retorne a quantidade de elementos negativos, nulos e positivos.

import java.util.Random;  
public class Quest4{  
 public static void main(String[] args){  
 int MAT[][] = new int[5][5];  
 int mat[] = Mat(MAT);  
 System.out.println("Quantidade de negativos " +mat[1]);   
 System.out.println("Quantidade de positivoss " +mat[0]);  
 System.out.println("Quantidade de nulos " +mat[2]);   
 }  
 public static int[] Mat(int MAT[][])  
 {  
 Random r = new Random();  
 int cont[] = new int[3];  
 for(int i = 0 ; i< MAT.length ; i++)  
 {  
 for(int j = 0 ; j< MAT[0].length ; j++)  
 {  
 MAT[i][j] = r.nextInt(10);  
 }  
 }  
   
   
 for(int i = 0 ; i< MAT.length ; i++)  
 {  
 for(int j = 0 ; j< MAT[0].length ; j++)  
 {  
 if(MAT[i][j] > 0)  
 cont[0]++;  
 else if(MAT[i][j] < 0)  
 cont[1]++;  
 else  
 cont[2]++;  
 }  
 }  
 return cont;   
  
   
 }  
}

5. Faça uma função que receba uma matriz de qualquer dimensão e faça sua linearização (transforme a matriz em um vetor).

import java.util.Random;  
public class Quest5{  
 public static void main(String[] args){  
 int MAT[][] = new int[2][2];  
 int vet[] = new int[MAT.length \* MAT[0].length];  
 vet = Vet(MAT);  
 for(int i = 0; i < MAT.length \* MAT[0].length; i++)  
 {  
 System.out.print(vet[i] + " ");  
 }   
   
 }  
   
 public static int[] Vet(int MAT[][])  
 {  
 Random r = new Random();  
 int VET[] = new int[MAT.length \* MAT[0].length];  
 int k = 0;  
 for(int i = 0 ; i< MAT.length ; i++)  
 {  
 for(int j = 0 ; j< MAT[0].length ; j++)  
 {  
   
 MAT[i][j] = r.nextInt(50);  
 }  
 }  
   
   
 for(int i = 0 ; i< MAT.length ; i++)  
 {  
 for(int j = 0 ; j< MAT[0].length ; j++)  
 {  
   
 VET[k] = MAT[i][j];  
 k++;  
 }  
 }   
   
 return VET;   
   
 }  
}

6. Faça uma função que receba um cubo qualquer dimensão e retorne a soma de suas faces.