



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

**Nombre:** Pedro Antonio Sanchez Sandoval  
Brayan Martin López Flores

**Carrera:** TSU en Tecnologías de la Información, Área Desarrollo de Software Multiplataforma. **Grupo:** 4ATI

**Asignatura:** Estructuras de Datos Aplicadas **Unidad:** Arreglos

**Profesor:** MGTI Jorge Luis De La Paz Ramos

1. Construya un programa que realice las siguientes acciones. Considere un arreglo ordenado en forma ascendente
  - Genere un arreglo de 100 números aleatorios entre 0 y 100
  - Dado un número ingresado por el usuario, de como resultado la posición de la primera ocurrencia.
  - Dado un número ingresado ver cuantas veces se repite en el arreglo.
  - Muestre el arreglo las veces que el usuario lo requiera.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Random rand = new Random();

int[] numeros = new int[100];
for (int i = 0; i < 100; i++) {
    numeros[i] = rand.nextInt(101);
}
Arrays.sort(numeros);

while (true) {
    System.out.println("1. Buscar numero");
    System.out.println("2. Ver arreglo");
    System.out.print("Opcion: ");
    int opcion = ent.nextInt();

    if (opcion == 1) {
        System.out.print("Número a buscar: ");
        int b = ent.nextInt();

        int posicion = -1;
        for (int i = 0; i < 100; i++) {
            if (numeros[i] == b) {
                posicion = i;
                break;
            }
        }
        int contador = 0;
        for (int i = 0; i < 100; i++) {
            if (numeros[i] == b) contador++;
        }

        if (posicion != -1) {
            System.out.println("Posicion: " + (posicion + 1));
            System.out.println("Se repite: " + contador + " veces");
        } else {
            System.out.println("No encontrado");
        }
    }
}
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
1. Buscar numero  
2. Ver arreglo  
Opcion: 3  
1. Buscar numero  
2. Ver arreglo  
Opcion: 6  
1. Buscar numero  
2. Ver arreglo  
Opcion: 2  
2  
2  
3  
5  
7  
7  
8  
8  
11  
12  
13  
13  
14  
14  
15  
18  
18  
19  
19  
19  
20  
20  
24  
24  
25  
26  
26  
27  
27
```

2. Una compañía distribuye N productos a distintos comercios de la ciudad. Para ello almacena en un arreglo toda la información relacionada con su mercancía:
- Clave
  - Descripción
  - Existencia
  - Mínimo para mantener de existencia.
  - Precio Unitario.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
---Inventario completo---
Producto 0
Clave: 6543 | Descripcion543 | Existencias 3 | Min 2 | Precio 12
Producto 1
Clave: 832 | Descripcion3432 | Existencias 3 | Min 1 | Precio 12
BUILD SUCCESSFUL (total time: 39 seconds)
```

```
/* Java Application logic here
Scanner ent = new Scanner(System.in);

System.out.println("Ingresa la cantidad de productos:");
int n = ent.nextInt();
int[] clave = new int[n];
String[] descripcion = new String[n];
int[] existencia = new int[n];
int[] mi = new int[n];
int[] pre = new int[n];

for(int i = 0; i<clave.length;i++){
    System.out.println("Ingresa la clave "+ i);
    clave[i] = ent.nextInt();
    System.out.println("Ingresa descripcion "+ i);
    descripcion[i] = ent.next();
    System.out.println("Ingresa la existencia "+ i);
    existencia[i] = ent.nextInt();
    System.out.println("Agrega el Minimo para mantener la existencia "+ i);
    mi[i] = ent.nextInt();
    System.out.println("Ingresa el precio unitario "+ i);
    pre[i] = ent.nextInt();
}
System.out.println("---Inventario completo---");
for(int i = 0; i<clave.length;i++){
    System.out.println("Producto "+ i);
    System.out.println("Clave: "+ clave[i]+ " | "+ "Descripcion"+descripcion[i]+ " | "+"Existencias "+existencia[i]+ " | "+"Min "+mi[i]+ " | "+"Pre"+pre[i]);
}
```

**3. Escriba un programa que efectúe las siguientes operaciones:**

- a) Venta de un producto: se deben actualizar los campos que correspondan y verificar que la nueva existencia no esté por debajo del mínimo. (Datos: clave, cantidad vendida)
- b) Reabastecimiento de un producto: se deben actualizar los campos que correspondan. (Datos: clave, cantidad comprada.)
- c) Actualizar el precio de un producto. (Datos: clave, porcentaje de aumento.)
- d) Informar sobre un productor: se deben proporcionar todos los datos relacionados con un producto. (Dato: clave)



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
Cuantos productos vas a tener? 2  
Ahora llena los datos de cada producto:  
Producto 1:  
Clave: 5432  
Nombre: leche  
Existencias: 56  
Minimo: 3  
Precio: 17  
Producto 2:  
Clave: 5302  
Nombre: papas  
Existencias: 60  
Minimo: 4  
Precio: 19  
Tienda  
1. Vender  
2. Reabastecimiento  
3. Cambiar precio  
4. Ver producto  
5. Salir  
Elige: 1  
Clave del producto: 5432  
Cudntos vender: 7  
Vendido. Quedan: 49  
Existencias: 49  
Precio: $17.0
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
double[] pre = new double[n];

System.out.println("Ahora llena los datos de cada producto:");
for(int i = 0; i < c.length; i++) {
    System.out.println("Producto " + (i + 1) + ":");

    System.out.print("Clave: ");
    c[i] = sc.nextInt();
    sc.nextLine();

    System.out.print("Nombre: ");
    no[i] = sc.nextLine();

    System.out.print("Existencias: ");
    ex[i] = sc.nextInt();

    System.out.print("Mínimo: ");
    mi[i] = sc.nextInt();

    System.out.print("Precio: ");
    pre[i] = sc.nextDouble();
}

System.out.println("Tienda");

while(true) {
    System.out.println("1. Vender");
    System.out.println("2. Reabastecimiento");
    System.out.println("3. Cambiar precio");
    System.out.println("4. Ver producto");
    System.out.println("5. Salir");

    System.out.print("Elige: ");
    int opcion = sc.nextInt();

    if(opcion == 4) break;

    System.out.print("Clave del producto: ");
    int claveBuscada = sc.nextInt();

    int pro = -1;
    for(int i = 0; i < c.length; i++) {
```

4. Al momento de su ingreso al hospital, a un paciente se le solicitan los siguientes datos:
- Nombre
  - Edad
  - Sexo
  - Domicilio:
    - Calle
    - Número
    - Ciudad
  - Teléfono
  - Seguro (este campo tendrá el valor VERDADERO si el paciente tiene seguro médico y FALSO en otro caso).
- Escriba un programa que pueda llevar a cabo las siguientes operaciones:
- Listar los nombres de todos los pacientes hospitalizados.
  - Obtener el porcentaje de pacientes hospitalizados en las siguientes categorías (dadas por la edad):



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

- a. Niños: hasta 13 años.
  - b. Jóvenes: mayores de 13 años y menores de 30
  - c. Adultos: mayores de 30 años.
- c) Obtener el porcentaje de hombres y de mujeres hospitalizados.
- d) Dado el nombre de un paciente, listar todos los datos relacionados con dicho paciente.

```
run:  
Ingresa la cantidad de pacientes 2  
Paciente 1:  
Nombre: joss  
Edad: 19  
Sexo (M/F): f  
tienes Seguro médico? (true/false): true  
Paciente 2:  
Nombre: kevin  
Edad: 17  
Sexo (M/F): m  
tienes Seguro médico? (true/false): false  
1. Listar pacientes  
2. Estadísticas edad  
3. Estadísticas sexo  
4. Buscar paciente  
5. Salir  
Opción: 5  
BUILD SUCCESSFUL (total time: 47 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
System.out.print("Ingresa la cantidad de pacientes ");
int n = ent.nextInt();
ent.nextLine();

String[] nom = new String[n];
int[] edades = new int[n];
String[] sexos = new String[n];
boolean[] seguros = new boolean[n];

for(int i = 0; i < n; i++) {
    System.out.println("Paciente " + (i + 1) + ":");
    System.out.print("Nombre: ");
    nom[i] = ent.nextLine();
    System.out.print("Edad: ");
    edades[i] = ent.nextInt();
    ent.nextLine();
    System.out.print("Sexo (M/F): ");
    sexos[i] = ent.nextLine();
    System.out.print("Tienes Seguro médico? (true/false): ");
    seguros[i] = ent.nextBoolean();
    ent.nextLine();
}

while(true) {
    System.out.println("1. Listar pacientes");
    System.out.println("2. Estadísticas edad");
    System.out.println("3. Estadísticas sexo");
    System.out.println("4. Buscar paciente");
    System.out.println("5. Salir");
    System.out.print("Opción: ");
    int op = ent.nextInt();
    ent.nextLine();

    if(op == 1) {
```

5. Supongamos que en una elección hubo 12 candidatos (con identificadores 1,2,3, ..., 12). Por otra parte, los votos para cada candidato se teclean de manera desorganizada como se muestra a continuación:

1 5 7 5 1 12 10 7 1 7 5 8 1 5 -1 (número negativo para detener la captura de votos)

Construya un programa que pueda proporcionar la siguiente información:

- El número de votos de cada candidato al final de la elección.
- El candidato ganador, el número de votos que obtuvo y el porcentaje correspondiente del total de la elección. Suponemos que el candidato ganador no empató en número de votos con otro candidato.



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
Ingresé los votos (1-12) y -1 para terminar:  
10  
-1  
--- RESULTADOS DE LA ELECCIÓN ---  
Candidato | Votos  
1 | 0  
2 | 0  
3 | 0  
4 | 0  
5 | 0  
6 | 0  
7 | 0  
8 | 0  
9 | 0  
10 | 1  
11 | 0  
12 | 0  
--- GANADOR ---  
Candidato: 10  
Votos: 1  
Porcentaje: 100.00%  
BUILD SUCCESSFUL (total time: 14 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Scanner ent = new Scanner (System.in);

int[] votos = new int[12];
int totalVotos = 0;

System.out.println("Ingrese los votos (1-12) y -1 para terminar:");

while(true) {
    int voto = ent.nextInt();

    if(voto == -1) {
        break; //
    }

    if(voto >= 1 && voto <= 12) {
        votos[voto - 1]++;
        totalVotos++;
    }
}

System.out.println("--- RESULTADOS DE LA ELECCIÓN ---");
System.out.println("Candidato | Votos");
for(int i = 0; i < 12; i++) {
    System.out.println(" " + (i + 1) + " | " + votos[i]);
}

int ganador = 0;
int maxVotos = votos[0];

for(int i = 1; i < 12; i++) {
    if(votos[i] > maxVotos) {
        maxVotos = votos[i];
        ganador = i;
    }
}
```

6. Almacene en un arreglo las temperaturas de un día (0-23 horas). Calcule su media e imprimir la temperatura más alta y la más baja; asimismo imprimir la hora respectiva. Por ejemplo, la temperatura media fue de 21.5 grados Celsius, la más alta de 29 grados Celsius a las 14 horas y la más baja de 9.4 grados Celsius a las 3 horas.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

run:

```
Ingrese las temperaturas para cada hora del dia:  
Temperatura a las 0 horas: 12  
Temperatura a las 1 horas: 12  
Temperatura a las 2 horas: 11  
Temperatura a las 3 horas: 7  
Temperatura a las 4 horas: 6  
Temperatura a las 5 horas: 3  
Temperatura a las 6 horas: 2  
Temperatura a las 7 horas: 3  
Temperatura a las 8 horas: 5  
Temperatura a las 9 horas: 8  
Temperatura a las 10 horas: 13  
Temperatura a las 11 horas: 16  
Temperatura a las 12 horas: 16  
Temperatura a las 13 horas: 15  
Temperatura a las 14 horas: 17  
Temperatura a las 15 horas: 19  
Temperatura a las 16 horas: 16  
Temperatura a las 17 horas: 15  
Temperatura a las 18 horas: 14  
Temperatura a las 19 horas: 12  
Temperatura a las 20 horas: 10  
Temperatura a las 21 horas: 7  
Temperatura a las 22 horas: 7  
Temperatura a las 23 horas: 6  
--- Resultados del dia ---  
Temperatura media: 10.5 C  
Temperatura más alta: 19.0 C a las 15 horas  
Temperatura más baja: 2.0 C a las 6 horas  
BUILD SUCCESSFUL (total time: 1 minute 32 seconds)
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here

    Scanner ent = new Scanner(System.in);
    double[] temp = new double[24];

    System.out.println("Ingrese las temperaturas para cada hora del dia:");

    // Leer temperaturas para cada hora
    for(int hora = 0; hora < 24; hora++) {
        System.out.print("Temperatura a las " + hora + " horas: ");
        temp[hora] = ent.nextDouble();
    }

    double suma = 0;
    for(int i = 0; i < 24; i++) {
        suma += temp[i];
    }
    double media = suma / 24;

    double maxTemp = temp[0];
    int horaMax = 0;

    for(int i = 1; i < 24; i++) {
        if(temp[i] > maxTemp) {
            maxTemp = temp[i];
            horaMax = i;
        }
    }

    double minTemp = temp[0];
    int horaMin = 0;

    for(int i = 1; i < 24; i++) {
        if(temp[i] < minTemp) {
            minTemp = temp[i];
        }
    }
```

7. Almacene en un arreglo la temperatura de cada día de una determinada semana y que realice lo siguiente:
- La temperatura promedio.
  - Un arreglo que contenga las diferencias de cada temperatura con respecto al promedio.
  - La menor temperatura y el número de día en que ocurrió.
  - La mayor temperatura y el número de día en que ocurrió.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
ingrese las temperaturas en grados Celsius:
```

```
Temperatura del Lunes: 17
Temperatura del Martes: 16
Temperatura del Miercoles: 19
Temperatura del Jueves: 20
Temperatura del Viernes: 21
Temperatura del Sabado: 15
Temperatura del Domingo: 17
```

```
---Resultados---
```

```
Temperaturas ingresadas:
Lunes      : 17.0 C
Martes     : 16.0 C
Miercoles  : 19.0 C
Jueves     : 20.0 C
Viernes    : 21.0 C
Sabado     : 15.0 C
Domingo    : 17.0 C
```

```
a. Temperatura promedio: 17.86 C
```

```
b. Diferencias con respecto al promedio:
```

```
Lunes      : -0.86 C
Martes     : -1.86 C
Miercoles  : 1.14 C
Jueves     : 2.14 C
Viernes    : 3.14 C
Sabado     : -2.86 C
Domingo    : -0.86 C
```

```
c. Menor temperatura: 15.0C
```

```
Dia: Sabado (Dia 6)
```

```
d. Mayor temperatura: 21.0C
```

```
Dia: Viernes (Dia 5)
```

```
BUILD SUCCESSFUL (total time: 28 seconds)
```

```
// TODO code application logic here

Scanner ent = new Scanner (System.in);
double[] temperaturas = new double[7];
String[] dias = {"Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"};

System.out.println("Ingrese las temperaturas de la semana ");
System.out.println("Ingrese las temperaturas en grados Celsius:\n");

for (int i = 0; i < 7; i++) {
    System.out.print("Temperatura del " + dias[i] + ": ");
    temperaturas[i] = ent.nextDouble();
}

double suma = 0;
for (double temp : temperaturas) {
    suma += temp;
}
double promedio = suma / temperaturas.length;

double[] diferencias = new double[7];
for (int i = 0; i < 7; i++) {
    diferencias[i] = temperaturas[i] - promedio;
}

double minTemp = temperaturas[0];
int diaMin = 0;
for (int i = 1; i < 7; i++) {
    if (temperaturas[i] < minTemp) {
        minTemp = temperaturas[i];
        diaMin = i;
    }
}

double maxTemp = temperaturas[0];
int diaMax = 0;
for (int i = 1; i < 7; i++) {
    if (temperaturas[i] > maxTemp) {
        maxTemp = temperaturas[i];
        diaMax = i;
    }
}
```

8. Almacene en un arreglo, 20 elementos de tipo entero e imprimir:



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

- a. La suma de los elementos que ocupan posiciones pares.
- b. El mayor de los elementos que ocupan posiciones impares.
- c. La posición del mayor número par.

```
run:  
Ingresе numero 1: 4  
Ingresе numero 2: 2  
Ingresе numero 3: 6  
Ingresе numero 4: 1  
Ingresе numero 5: 9  
Ingresе numero 6: 3  
Ingresе numero 7: 67  
Ingresе numero 8: 34  
Ingresе numero 9: 21  
Ingresе numero 10: 12  
Ingresе numero 11: 8  
Ingresе numero 12: 80  
Ingresе numero 13: 9  
Ingresе numero 14: 4  
Ingresе numero 15: 7  
Ingresе numero 16: 9  
Ingresе numero 17: 1  
Ingresе numero 18: 5  
Ingresе numero 19: 5  
Ingresе numero 20: 1  
a. Suma posiciones pares: 137  
b. Mayor en posiciones impares: 80  
c. Mayor numero par: 80 en posicion 11  
BUILD SUCCESSFUL (total time: 25 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
/*
public static void main(String[] args) {
    // TODO code application logic here

    Scanner scanner = new Scanner(System.in);
    int[] n = new int[20];

    for (int i = 0; i < 20; i++) {
        System.out.print("Ingrese numero " + (i + 1) + ": ");
        n[i] = scanner.nextInt();
    }

    int sumaPares = 0;
    for (int i = 0; i < 20; i += 2) {
        sumaPares += n[i];
    }

    int mi = n[1];
    for (int i = 3; i < 20; i += 2) {
        if (n[i] > mi) {
            mi = n[i];
        }
    }

    int mp = -1;
    int posicionMayorPar = -1;
    for (int i = 0; i < 20; i++) {
        if (n[i] % 2 == 0) {
            if (mp == -1 || n[i] > mp) {
                mp = n[i];
                posicionMayorPar = i;
            }
        }
    }
}
```

9. Almacene en un arreglo, n elementos (máximo 30) e imprimir la suma:

- a. de números pares.
- b. de números impares.
- c. total de los elementos del arreglo.

```
run:
Ingresa la cantidad (maximo 30): 2
Número 1: 34
Número 2: 65
    Suma de numeros pares: 34
    Suma de numeros impares: 65
    Suma total: 99
BUILD SUCCESSFUL (total time: 10 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here

    Scanner scanner = new Scanner(System.in);

    System.out.print("Ingresa la cantidad (maximo 30): ");
    int n = scanner.nextInt();

    if (n <= 0 || n > 30) {
        System.out.println("debe ser entre 1 y 30");
        return;
    }

    int[] nu = new int[n];

    for (int i = 0; i < n; i++) {
        System.out.print("Número " + (i + 1) + ": ");
        nu[i] = scanner.nextInt();
    }

    int sp = 0;
    int si = 0;
    int st = 0;

    for (int i = 0; i < n; i++) {
        st += nu[i];

        if (nu[i] % 2 == 0) {
            sp += nu[i];
        } else {
            si += nu[i];
        }
    }
    System.out.println(" Suma de numeros pares: " + sp);
```

10. Almacene en un arreglo 15 números e imprimir cuántos son ceros, cuántos son negativos y cuántos positivos. Imprimir además la suma de los negativos y la suma de los positivos.



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
Ingrese 15 números:  
Número 1: 4  
Número 2: 8  
Número 3: 2  
Número 4: 9  
Número 5: 5  
Número 6: 1  
Número 7: 9  
Número 8: 4  
Número 9: 7  
Número 10: 2  
Número 11: 8  
Número 12: 4  
Número 13: 8  
Número 14: 2  
Número 15: 7  
Resultados:  
Ceros: 0  
Negativos: 0  
Positivos: 15  
Suma de negativos: 0  
Suma de positivos: 80  
BUILD SUCCESSFUL (total time: 12 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here

    Scanner ent = new Scanner(System.in);
    int[] numeros = new int[15];

    System.out.println("Ingrese 15 números:");

    for (int i = 0; i < 15; i++) {
        System.out.print("Número " + (i + 1) + ": ");
        numeros[i] = ent.nextInt();
    }

    int c = 0;
    int n = 0;
    int p = 0;
    int sn = 0;
    int sp = 0;

    for (int i = 0; i < 15; i++) {
        if (numeros[i] == 0) {
            c++;
        } else if (numeros[i] < 0) {
            n++;
            sn += numeros[i];
        } else {
            p++;
            sp += numeros[i];
        }
    }
    System.out.println("Resultados:");
    System.out.println("Ceros: " + c);
    System.out.println("Negativos: " + n);
    System.out.println("Positivos: " + p);
    System.out.println("Suma de negativos: " + sn);
```

11. Inicialice un arreglo con 10 valores. Genere un arreglo a con los números negativos y otro arreglo b con los positivos o iguales a cero.

```
run:
Arreglo original:
5 -3 8 -12 0 15 -7 22 -1 9

Arreglo a (negativos):
-3 -12 -7 -1

Arreglo b (positivos o cero):
5 8 0 15 22 9 BUILD SUCCESSFUL (total time: 0 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Scanner ent = new Scanner(System.in);

int[] numeros = {5, -3, 8, -12, 0, 15, -7, 22, -1, 9};

int cantNegativos = 0;
int cantPositivos = 0;

for(int i = 0; i < numeros.length; i++) {
    if(numeros[i] < 0) {
        cantNegativos++;
    } else {
        cantPositivos++;
    }
}

int[] a = new int[cantNegativos];
int[] b = new int[cantPositivos];

int indiceA = 0;
int indiceB = 0;

for(int i = 0; i < numeros.length; i++) {
    if(numeros[i] < 0) {
        a[indiceA] = numeros[i];
        indiceA++;
    } else {
        b[indiceB] = numeros[i];
        indiceB++;
    }
}
```

12. Almacene en un arreglo n elementos de tipo real (máximo 25) e imprima los siguientes valores:

- Máximo.
- Mínimo.
- La media de los que estén en posiciones pares.
- La varianza. (<http://www.disfrutalasmaticas.com/datos/desviacion-estandar.html>)



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:
Cuantos numeros desea ingresar? (maximo 25): 2

Ingrese los numeros:
Numero 1: 54
Numero 2: 23
Resultado
Maximo: 54.0
Minimo: 23.0
Media de posiciones pares: 54.0
Varianza: 240.25
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    int n;
    do {
        System.out.print("Cuantos numeros desea ingresar? (maximo 25): ");
        n = ent.nextInt();
    } while(n < 1 || n > 25);

    double[] numeros = new double[n];

    System.out.println("\nIngrese los numeros:");
    for(int i = 0; i < n; i++) {
        System.out.print("Número " + (i+1) + ": ");
        numeros[i] = ent.nextDouble();
    }

    double maximo = numeros[0];
    for(int i = 1; i < n; i++) {
        if(numeros[i] > maximo) {
            maximo = numeros[i];
        }
    }

    double minimo = numeros[0];
    for(int i = 1; i < n; i++) {
        if(numeros[i] < minimo) {
            minimo = numeros[i];
        }
    }
}
```

13. Almacene en un arreglo 10 números y cambie algún número por otro del mismo arreglo, realice el cambio y muestre el arreglo modificado.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

run:

Ingrese 10 numeros:

Numero 1: 4

Numero 2: 8

Numero 3: 2

Numero 4: 7

Numero 5: 3

Numero 6: 9

Numero 7: 1

Numero 8: 6

Numero 9: 8

Numero 10: 3

Arreglo original

Posicion: 0 1 2 3 4 5 6 7 8 9

Valor: 4 8 2 7 3 9 1 6 8 3

Qu~~d~~ n~~d~~meros desea intercambiar?

Primera posicion (0-9): 5

Segunda posicion (0-9): 9

Arreglo editado

Posicion: 0 1 2 3 4 5 6 7 8 9

Valor: 4 8 2 7 3 3 1 6 8 9

Se intercambio el numero en posicion 5 con el de posicion 9

BUILD SUCCESSFUL (total time: 17 seconds)



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    int[] numeros = new int[10];

    System.out.println("Ingrese 10 números:");
    for(int i = 0; i < 10; i++) {
        System.out.print("Número " + (i+1) + ": ");
        numeros[i] = ent.nextInt();
    }

    System.out.println("\nArreglo original");
    System.out.print("Posición: ");
    for(int i = 0; i < 10; i++) {
        System.out.print(i + " ");
    }
    System.out.print("\nValor: ");
    for(int i = 0; i < 10; i++) {
        System.out.print(numeros[i] + " ");
    }

    System.out.println("\n\nQué números desea intercambiar?");
    int pos1, pos2;

    do {
        System.out.print("Primera posición (0-9): ");
        pos1 = ent.nextInt();
        pos2 = ent.nextInt();
    } while(pos1 < 0 || pos1 > 9);
    do {
```

14. Genere aleatoriamente un arreglo de tamaño 20 con números comprendidos entre 1 y 20 e imprimir:
- Suma de los elementos.
  - Promedio de los elementos.
  - Varianza y desviación estándar de los valores.
  - Elemento que más se repite (moda).



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
Arreglo generado:  
5  
6 7  
8 8  
9 4  
5 32  
63 7  
8 6  
7 32  
63 8  
9 5  
6 3  
4 2  
3 7  
8 5  
6 3  
4  
5  
6 8  
9 45  
86 76  
147 32  
63 Resultado  
a) Suma de los elementos: 520  
b) Promedio: 26.0  
c) Varianza: 1404.3  
    Desviación estandar: 37.473990980412  
d) Moda (elemento que más se repite): 6  
    (Se repite 4 veces)  
BUILD SUCCESSFUL (total time: 20 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    int[] numeros = new int[20];

    System.out.println("Arreglo generado:");
    for (int i = 0; i < 20; i++) {
        numeros[i] = ent.nextInt(20) + 1;
        System.out.print(numeros[i] + " ");
    }

    int suma = 0;
    for (int i = 0; i < 20; i++) {
        suma += numeros[i];
    }

    double promedio = (double) suma / 20;

    double sumaCuadrados = 0;
    for (int i = 0; i < 20; i++) {
        sumaCuadrados += Math.pow(numeros[i] - promedio, 2);
    }
    double varianza = sumaCuadrados / 20;
    double desviacion = Math.sqrt(varianza);

    int moda = numeros[0];
    int maxRepeticiones = 0;
```

15. Almacene en un arreglo de tamaño 10 números reales. Calcule el promedio e indique cuántos elementos del arreglo son mayores que el promedio y genere otro arreglo con los menores o iguales.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:
Ingrese 10 numeros reales:
Número 1: 3
Número 2: 8
Número 3: 1
Número 4: 6
Número 5: 9
Número 6: 6
Número 7: 4
Número 8: 9
Número 9: 2
Número 10: 5
Resultados

Arreglo original:
3.0 8.0 1.0 6.0 9.0 6.0 4.0 9.0 2.0 5.0

Promedio: 5.3

Cantidad de elementos mayores que el promedio: 5

Arreglo con elementos menores o iguales al promedio:
3.0 1.0 4.0 2.0 5.0
BUILD SUCCESSFUL (total time: 10 seconds)
```

```
double[] numeros = new double[10];

// Llenar el arreglo
System.out.println("Ingrese 10 numeros reales:");
for(int i = 0; i < 10; i++) {
    System.out.print("Número " + (i+1) + ": ");
    numeros[i] = ent.nextDouble();
}

double suma = 0;
for(int i = 0; i < 10; i++) {
    suma += numeros[i];
}
double promedio = suma / 10;

int cantMayores = 0;
int cantMenoresIguales = 0;

for(int i = 0; i < 10; i++) {
    if(numeros[i] > promedio) {
        cantMayores++;
    } else {
        cantMenoresIguales++;
    }
}

double[] menoresIguales = new double[cantMenoresIguales];
int indice = 0;

for(int i = 0; i < 10; i++) {
    if(numeros[i] <= promedio) {
        menoresIguales[indice] = numeros[i];
        indice++;
    }
}
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

16. Almacene en un arreglo de n números enteros (máximo 30) y determine cuántos números terminan en dígito primo.

```
run:
Ingresé la cantidad de numeros (maximo 30): 3
Ingresé los 3 numeros:
Número 1: 4
Número 2: 7
Número 3: 9

Numeros que terminan en digito primo (2, 3, 5, 7):
7 termina en 7

Total de numeros que terminan en digito primo: 1
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    System.out.print("Ingresé la cantidad de numeros (maximo 30): ");
    int n = ent.nextInt();

    if (n > 30 || n <= 0) {
        System.out.println("Error: Debe ingresar un valor entre 1 y 30");
        ent.close();
        return;
    }

    int[] numeros = new int[n];

    System.out.println("Ingresé los " + n + " numeros:");
    for (int i = 0; i < n; i++) {
        System.out.print("Número " + (i + 1) + ": ");
        numeros[i] = ent.nextInt();
    }

    int contador = 0;
    System.out.println("\nNumeros que terminan en digito primo (2, 3, 5, 7):");

    for (int i = 0; i < n; i++) {

        int ultimoDigito = Math.abs(numeros[i]) % 10;

        if (ultimoDigito == 2 || ultimoDigito == 3 ||
            ultimoDigito == 5 || ultimoDigito == 7) {
            System.out.println(numeros[i] + " termina en " + ultimoDigito);
            contador++;
        }
    }
}
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

17. Convierta un número entero decimal a su equivalente en representación binaria. (

```
run:
Ingresé un numero decimal: 10.75
El numero 10.75 en binario es: 1010.11
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
Scanner ent = new Scanner(System.in);

System.out.print("Ingresé un numero decimal: ");
double decimal = ent.nextDouble();
| 

int parteEntera = (int) decimal;
double parteDecimal = decimal - parteEntera;

int[] binarioEntero = new int[32];
int indice = 0;

if (parteEntera == 0) {
    binarioEntero[indice] = 0;
    indice = 1;
} else {
    int temp = parteEntera;
    while (temp > 0) {
        binarioEntero[indice] = temp % 2;
        temp = temp / 2;
        indice++;
    }
}

double[] binarioDecimal = new double[32];
int indiceDecimal = 0;
int maxIteraciones = 10;

while (parteDecimal > 0 && indiceDecimal < maxIteraciones) {
    parteDecimal = parteDecimal * 2;
    if (parteDecimal >= 1) {
        binarioDecimal[indiceDecimal] = 1;
        parteDecimal = parteDecimal - 1;
    } else {
        binarioDecimal[indiceDecimal] = 0;
    }
}
```

18. Almacene en un arreglo n números enteros y determine ¿cuántas veces se repite cada uno de ellos?

Si  $n=6$  y los elementos del arreglo son: 3,4,6,6,4,6, se imprimirá:  
3=1  
4=2  
6=3



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:
Ingresé el tamaño del arreglo (n): 3
Ingresé los 3 elementos del arreglo:
Elemento 1: 4
Elemento 2: 9
Elemento 3: 1

Resultados:
4 = 1
9 = 1
1 = 1
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    System.out.print("Ingresé el tamaño del arreglo (n): ");
    int n = ent.nextInt();

    int[] numeros = new int[n];

    System.out.println("Ingresé los " + n + " elementos del arreglo:");
    for (int i = 0; i < n; i++) {
        System.out.print("Elemento " + (i + 1) + ": ");
        numeros[i] = ent.nextInt();
    }

    boolean[] yaContado = new boolean[n];

    System.out.println("\nResultados:");
    for (int i = 0; i < n; i++) {

        if (yaContado[i]) {
            continue;
        }

        int contador = 1;

        for (int j = i + 1; j < n; j++) {
            if (numeros[i] == numeros[j]) {
                contador++;
                yaContado[j] = true;
            }
        }
    }
}
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

19. Almacene en un arreglo  $n$  números ( $n$  debe ser un numero par, y como máximo 50), valide si el tamaño es par e invertir los elementos a la mitad del arreglo. Imprimir el vector original y el invertido.

Si  $n=6$ ,  $v=[1][2][3][4][5][6]$   $v(\text{invertido})=[3][2][1][6][5][4]$ .

```
run:  
Ingrese el tamano del arreglo (debe ser par, maximo 50): 2  
Ingrese los 2 numeros:  
Numero 1: 7  
Numero 2: 10  
  
Arreglo original:  
7 10  
  
Arreglo invertido (mitades intercambiadas):  
10 7  
BUILD SUCCESSFUL (total time: 9 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    System.out.print("Ingrese el tamano del arreglo (debe ser par, maximo 50): ");
    int n = ent.nextInt();

    if (n % 2 != 0) {
        System.out.println("Error: El tamano debe ser un numero par");
        ent.close();
        return;
    }
    if (n > 50 || n <= 0) {
        System.out.println("Error: El tamano debe estar entre 1 y 50");
        ent.close();
        return;
    }

    int[] numeros = new int[n];

    System.out.println("Ingrese los " + n + " numeros:");
    for (int i = 0; i < n; i++) {
        System.out.print("Número " + (i + 1) + ": ");
        numeros[i] = ent.nextInt();
    }

    System.out.println("\nArreglo original:");
    for (int i = 0; i < n; i++) {
        System.out.print(numeros[i] + " ");
    }
    System.out.println();
```

- 20.** Almacene en dos arreglos números reales a y b de 10 elementos cada uno, a partir de ellos genere un tercer arreglo con el siguiente criterio: sumar el primer elemento de a más el último elemento de b y luego el segundo elemento de a con el noveno elemento de b y así sucesivamente hasta llegar al décimo elemento de a más el primer elemento de b. Imprimir las sumas almacenadas en el arreglo c.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
Ingresese los 10 elementos del arreglo A:  
A[0]: 4  
A[1]: 5  
A[2]: 2  
A[3]: 7  
A[4]: 8  
A[5]: 1  
A[6]: 9  
A[7]: 54  
A[8]: 8  
A[9]: 3  
  
Ingresese los 10 elementos del arreglo B:  
B[0]: 876  
B[1]: 7  
B[2]: 98  
B[3]: 23  
B[4]: 78  
B[5]: 45  
B[6]: 65  
B[7]: 98  
B[8]: 32  
B[9]: 65  
  
Arreglo C (sumas resultantes):  
C[0] = A[0] + B[9] = 4.0 + 65.0 = 69.0  
C[1] = A[1] + B[8] = 5.0 + 32.0 = 37.0  
C[2] = A[2] + B[7] = 2.0 + 98.0 = 100.0  
C[3] = A[3] + B[6] = 7.0 + 65.0 = 72.0  
C[4] = A[4] + B[5] = 8.0 + 45.0 = 53.0  
C[5] = A[5] + B[4] = 1.0 + 78.0 = 79.0  
C[6] = A[6] + B[3] = 9.0 + 23.0 = 32.0  
C[7] = A[7] + B[2] = 54.0 + 98.0 = 152.0  
C[8] = A[8] + B[1] = 8.0 + 7.0 = 15.0  
C[9] = A[9] + B[0] = 3.0 + 876.0 = 879.0  
BUILD SUCCESSFUL (total time: 46 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here
    Scanner ent = new Scanner(System.in);

    double[] a = new double[10];
    double[] b = new double[10];
    double[] c = new double[10];

    System.out.println("Ingrese los 10 elementos del arreglo A:");
    for (int i = 0; i < 10; i++) {
        System.out.print("A[" + i + "]: ");
        a[i] = ent.nextDouble();
    }

    System.out.println("\nIngrese los 10 elementos del arreglo B:");
    for (int i = 0; i < 10; i++) {
        System.out.print("B[" + i + "]: ");
        b[i] = ent.nextDouble();
    }

    for (int i = 0; i < 10; i++) {
        c[i] = a[i] + b[9 - i];
    }

    System.out.println("\nArreglo C (sumas resultantes):");
    for (int i = 0; i < 10; i++) {
        System.out.println("C[" + i + "] = A[" + i + "] + B[" + (9 - i) + "] = "
                           + a[i] + " + " + b[9 - i] + " = " + c[i]);
    }
}
```

21. Almacene un arreglo de longitud n ordenado en forma ascendente y un elemento x del mismo tipo que los elementos del arreglo, intercalar x en el arreglo v de manera que siga ordenado.

```
run:
Ingresa la longitud : 2
Ingresa los elementos del arreglo:
2
2
Ingresa el elemento: 34

Arreglo con el elemento:
2 2 34 BUILD SUCCESSFUL (total time: 12 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
System.out.print("Ingresa la longitud : ");
int n = ent.nextInt();

int[] arreglo = new int[n];

System.out.println("Ingresa los elementos del arreglo:");
for (int i = 0; i < n; i++) {
    arreglo[i] = ent.nextInt();
}

System.out.print("Ingresa el elemento: ");
int x = ent.nextInt();

int[] nuevoArreglo = new int[n + 1];
boolean insertado = false;
int j = 0;

for (int i = 0; i < n; i++) {
    if (!insertado && x < arreglo[i]) {
        nuevoArreglo[j++] = x;
        insertado = true;
    }
    nuevoArreglo[j++] = arreglo[i];
}

if (!insertado) {
    nuevoArreglo[j] = x;
}

System.out.println("\nArreglo con el elemento:");
for (int i = 0; i < nuevoArreglo.length; i++) {
    System.out.print(nuevoArreglo[i] + " ");
}
```

22. Almacene 2 arreglos a y b de longitudes n y m, el primero ordenado en forma ascendente y el segundo ordenado de manera descendente, crear un nuevo arreglo c de n + m elementos intercalando los elementos de a y b de modo que c quede ordenado en forma ascendente.

```
run:
Ingresa la longitud del arreglo A: 3
Ingresa los elementos del arreglo A :
5
8
9
Ingresa la longitud del arreglo B: 2
Ingresa los elementos del arreglo B):
6
9

Arreglo C en orden:
5 8 9 9 6 BUILD SUCCESSFUL (total time: 13 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Scanner ent = new Scanner(System.in);

System.out.print("Ingresa la longitud del arreglo A: ");
int n = ent.nextInt();
int[] a = new int[n];

System.out.println("Ingresa los elementos del arreglo A :");
for (int i = 0; i < n; i++) {
    a[i] = ent.nextInt();
}

System.out.print("Ingresa la longitud del arreglo B: ");
int m = ent.nextInt();
int[] b = new int[m];

System.out.println("Ingresa los elementos del arreglo B:");
for (int i = 0; i < m; i++) {
    b[i] = ent.nextInt();
}

for (int i = 0; i < m / 2; i++) {
    int temp = b[i];
    b[i] = b[m - 1 - i];
    b[m - 1 - i] = temp;
}

int[] c = new int[n + m];
int i = 0, j = 0, k = 0;

while (i < n && j < m) {
    if (a[i] <= b[j]) {
        c[k++] = a[i++];
    } else {
        c[k++] = b[j++];
    }
}
```

23. Calcular la suma de todos los elementos de un arreglo.

```
run:
Ingrese el tamano del arreglo: 2
elemento 1: 5
elemento 2: 9

la suma de los elementos es: 14
BUILD SUCCESSFUL (total time: 5 seconds)
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
Scanner ent = new Scanner(System.in);

int suma = 0;

System.out.print("Ingrese el tamano del arreglo: ");
int n = ent.nextInt();

int[] arreglo = new int[n];
for (int i = 0; i < n; i++) {
    System.out.print("elemento " + (i + 1) + ": ");
    arreglo[i] = ent.nextInt();
    suma += arreglo[i];
}

System.out.println("\nla suma de los elementos es: " + suma);
```

24. Realizar un arreglo que registre n palabras y muestre aquellas que inician con una vocal.

```
run:
Ingrese la cantidad de palabras: 2
Ingresa las palabras:
oso
estrella

inicia con una vocal:
oso
estrella
BUILD SUCCESSFUL (total time: 11 seconds)
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
System.out.print("Ingrese la cantidad de palabras: ");
int n = ent.nextInt();
ent.nextLine();

String[] palabras = new String[n];

System.out.println("Ingresa las palabras:");
for (int i = 0; i < n; i++) {
    palabras[i] = ent.nextLine();
}

System.out.println("\n inicia con una vocal:");
for (int i = 0; i < n; i++) {
    String palabra = palabras[i];

    if (palabra.length() > 0) {
        char primera = palabra.charAt(0);

        if (primera == 'a' || primera == 'A' ||
            primera == 'e' || primera == 'E' ||
            primera == 'i' || primera == 'I' ||
            primera == 'o' || primera == 'O' ||
            primera == 'u' || primera == 'U') {

            System.out.println(palabra);
        }
    }
}

ent.close();
```

25. Escriba un arreglo que solicite las letras del alfabeto y las imprima de manera inversa.

```
run:
Ingresa el numero de letras del alfabeto: 4
Ingresa la letra del alfabeto1: a
Ingresa la letra del alfabeto2: b
Ingresa la letra del alfabeto3: c
Ingresa la letra del alfabeto4: d

orden inverso:
d c b a
BUILD SUCCESSFUL (total time: 10 seconds)
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {  
    // TODO code application logic here  
  
    Scanner ent = new Scanner(System.in);  
  
    System.out.print("Ingresa el numero de letras del alfabeto: ");  
    int n = ent.nextInt();  
  
    String[] letras = new String[n];  
    for (int i = 0; i < n; i++) {  
        System.out.print("Ingresa la letra del alfabeto" + (i + 1) + ": ");  
        letras[i] = ent.next();  
    }  
    System.out.println("\n orden inverso:");  
    for (int i = n - 1; i >= 0; i--) {  
        System.out.print(letras[i] + " ");  
    }  
  
    System.out.println();  
}
```

- 26.** Una aplicación de las computadoras es dibujar gráficos de barra(llamados histogramas). Escriba un programa que dibuje un gráfico como el siguiente donde la cantidad de asteriscos de cada línea corresponda al número que está a la par.



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

3. | \*\*\*
  5. | \*\*\*\*\*
  7. | \*\*\*\*\*
  9. | \*\*\*\*\*
- 

```
Ingresá el número 15: 3
Ingresá el número 16: 8
Ingresá el número 17: 3
Ingresá el número 18: 6
Ingresá el número 19: 9
Ingresá el número 20: 8
Ingresá el número 21: 1
Ingresá el número 22: 2
Ingresá el número 23: 5
Ingresá el número 24: 4
```

histograma:

```
6 | *****
9 | *****
3 | ***
2 | **
7 | *****
4 | ****
7 | *****
2 | **
7 | *****
9 | *****
3 | ***
6 | *****
1 | *
4 | ****
3 | ***
8 | *****
3 | ***
6 | *****
9 | *****
8 | *****
1 | *
2 | **
5 | ****
4 | ****
```

```
BUILD SUCCESSFUL (total time: 21 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public class arreglo_26 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
  
        Scanner ent = new Scanner(System.in);  
  
        System.out.print("cuantos numeros deseas ingresar ");  
        int cantidad = ent.nextInt();  
  
        int[] numeros = new int[cantidad];  
  
        for (int i = 0; i < cantidad; i++) {  
            System.out.print("Ingresa el numero " + (i + 1) + ": ");  
            numeros[i] = ent.nextInt();  
        }  
  
        System.out.println("\nhistograma:");  
  
        for (int i = 0; i < cantidad; i++) {  
            System.out.print(numeros[i] + " | ");  
            for (int j = 0; j < numeros[i]; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

27. Escribir un programa que visualice la siguiente salida:

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3  
1 2  
1
```



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3  
1 2  
1  
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
public class arreglo_27 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
  
        for (int i = 1; i <= 4; i++) {  
            for (int j = 1; j <= i; j++) {  
                System.out.print(j + " ");  
            }  
            System.out.println();  
        }  
  
        for (int i = 3; i >= 1; i--) {  
            for (int j = 1; j <= i; j++) {  
                System.out.print(j + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

**28.** Escriba un programa que imprima la forma en diamante siguiente.

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



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:
 *
 ***
 *****
 *****
*****
 *****
 *****
 ***
 *
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
/*
public static void main(String[] args) {
    // TODO code application logic here

    int n = 5;

    for (int i = 1; i <= n; i++) {
        for (int j = i; j < n; j++) {
            System.out.print(" ");
        }
        for (int j = 1; j <= (2 * i - 1); j++) {
            System.out.print("*");
        }
        System.out.println();
    }

    for (int i = n - 1; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            System.out.print(" ");
        }
        for (int j = 1; j <= (2 * i - 1); j++) {
            System.out.print("*");
        }
        System.out.println();
    }
}
```

29. Escriba un programa que imprima los siguientes patrones por separado, uno debajo del siguiente. Utilice ciclos “for” para generar los patrones.



# **UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA**

(A)	(B)	(C)	(D)
*	*****	*****	*
**	*****	*****	***
***	*****	*****	***
****	*****	*****	****
*****	*****	*****	*****
*****	****	****	*****
*****	***	***	*****
*****	**	**	*****
*****	*	*	*****



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print("*");
    }
    System.out.println();
}

System.out.println();

for (int i = n; i >= 1; i--) {
    for (int j = 1; j <= i; j++) {
        System.out.print("*");
    }
    System.out.println();
}

System.out.println();

for (int i = n; i >= 1; i--) {
    for (int j = n; j > i; j--) {
        System.out.print(" ");
    }
    for (int j = 1; j <= i; j++) {
        System.out.print("*");
    }
    System.out.println();
}

System.out.println();

for (int i = 1; i <= n; i++) {
    for (int j = i; j < n; j++) {
        System.out.print(" ");
    }
    for (int j = 1; j <= i; j++) {
```

### 30. Imprimir las siguientes series

1 × 9+2 = 11  
12 × 9+3 = 111  
123 × 9+4 = 1111  
1234 × 9+5 = 11111  
12345 × 9+6 = 111111  
123456 × 9+7 = 1111111  
1234567 × 9+8 = 11111111  
12345678 × 9+9 = 111111111  
123456789 × 9+10 = 1111111111

1  
1 × 1 = 1  
11 × 11 = 121  
111 × 111 = 12321  
1111 × 1111 = 1234321  
11111 × 11111 = 123454321  
111111 × 111111 = 12345654321  
1111111 × 1111111 = 1234567654321  
11111111 × 11111111 = 123456787654321



## UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
run:  
S 1:  
1 x 9 + 2 = 11  
12 x 9 + 3 = 111  
123 x 9 + 4 = 1111  
1234 x 9 + 5 = 11111  
12345 x 9 + 6 = 111111  
123456 x 9 + 7 = 1111111  
1234567 x 9 + 8 = 11111111  
12345678 x 9 + 9 = 111111111  
123456789 x 9 + 10 = 1111111111  
  
S 2:  
1 x 1 = 1  
11 x 11 = 121  
111 x 111 = 12321  
1111 x 1111 = 1234321  
11111 x 11111 = 123454321  
111111 x 111111 = 12345654321  
1111111 x 1111111 = 1234567654321  
11111111 x 11111111 = 123456787654321  
BUILD SUCCESSFUL (total time: 0 seconds)
```



# UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

```
public static void main(String[] args) {
    // TODO code application logic here

    System.out.println("S 1:");
    int num1 = 0;
    for (int i = 1; i <= 9; i++) {
        num1 = num1 * 10 + i;
        int resultado = num1 * 9 + (i + 1);
        System.out.println(num1 + " x 9 + " + (i + 1) + " = " + resultado);
    }

    System.out.println("\nS 2:");

    for (int i = 1; i <= 9; i++) {
        long n = 0;
        for (int j = 1; j <= i; j++) {
            n = n * 10 + 1;
        }
        long resultado = n * n;
        System.out.println(n + " x " + n + " = " + resultado);
    }
}
```