

1. Si se tiene un vector con n elementos. hallar el promedio de los números impares

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; public class Ejer1 { ////////////////////////////////////// ////////////////////////////////////// public static double promedioImpares(int n, int a[]) { int suma = 0; int contador = 0; for (int i = 0; i < n; i++) { if (a[i] % 2 != 0) { // Verifica si el número es impar suma += a[i]; contador++; } } if (contador == 0) { return 0; } else { return (double) suma / contador; } } public static void LlenaV(int n,int A[]){ Scanner lee=new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("V["+i+""]=""); A[i]=lee.nextInt(); } } public static void MostrarV(int m,int b[]){ for (int i = 0; i < m; i++) { System.out.print(" "+b[i]); } } ////////////////////////////////////// ////////////////////////////////////// public static void main(String[] args) { int n; Scanner lee = new Scanner(System.in); System.out.print("cantidad de elementos: "); n = lee.nextInt(); int a[] = new int[n]; LlenaV(n, a); MostrarV(n, a); double promedio = promedioImpares(n, a); System.out.println("\nPromedio de números impares: " + promedio); } </pre>

	}
--	---

Corrida:

```
cantidad de elementos: 5
V[0]=1
V[1]=2
V[2]=3
V[3]=4
V[4]=5
|1|2|3|4|5
Promedio de números impares: 3.0
```

2. Si se tiene un vector con n elementos. Halla la suma de los elementos que ocupan las posiciones impares.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; public class Ejer2 { //Si se tiene un vector con n elementos. Halla la suma de los elementos que ocupan las posiciones impares. public static int sumaPosicionesImpares(int n, int a[]) { int suma = 0; for (int i = 1; i < n; i += 2) { suma += a[i]; } return suma; } public static void LlenaV(int n,int A[]){ Scanner lee=new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("V["+i+"]"+"="); A[i]=lee.nextInt(); } public static void MostrarV(int m,int b[]){ for (int i = 0; i < m; i++) { System.out.print(" "+b[i]); } } //////////////////////////////////// //////////////////////////////////// public static void main(String[] args) { int n; Scanner lee = new Scanner(System.in); System.out.print("cantidad de elementos: "); n = lee.nextInt(); int a[] = new int[n]; LlenaV(n, a); MostrarV(n, a); int suma = sumaPosicionesImpares(n, a); System.out.println("\nSuma de elementos en posiciones impares: " + suma); }} </pre>

Corrida:


```
////////////////////////////////////  
public static void main(String[] args) {  
    int n;  
    Scanner lee = new Scanner(System.in);  
    System.out.print("cantidad de elementos: ");  
    n = lee.nextInt();  
    int a[] = new int[n];  
  
    LlenaV(n, a);  
    MostrarV(n, a);  
  
    mayorYMenor(n, a);  
}}
```

Corrida:

```
run:  
cantidad de elementos: 4  
V[0]=1  
V[1]=2  
V[2]=3  
V[3]=4  
|1|2|3|4El mayor elemento es: 4  
El menor elemento es: 1
```

4. Si se tiene un vector con n elementos. Determinar cuántos números primos hay en el vector.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer4 { //Si se tiene un vector con n elementos. Determinar cuántos números primos hay en el vector. public static boolean esPrimo(int num) { if (num <= 1) { return false; } for (int i = 2; i <= Math.sqrt(num); i++) { if (num % i == 0) { return false; } } return true; } public static int contarPrimos(int n, int a[]) { int contadorPrimos = 0; for (int i = 0; i < n; i++) { if (esPrimo(a[i])) { contadorPrimos++; } } return contadorPrimos; } public static void LlenaV(int n,int A[]){ Scanner lee=new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("V["+i+"]"+"="); A[i]=lee.nextInt(); } public static void MostrarV(int m,int b[]){ for (int i = 0; i < m; i++) { System.out.print(" "+b[i]); } } ////////////////////////////////////// ////////////////////////////////////// public static void main(String[] args) { int n; Scanner lee = new Scanner(System.in); System.out.print("cantidad de elementos: "); n = lee.nextInt(); int a[] = new int[n]; </pre>

```

        LlenaV(n, a);
        MostrarV(n, a);

        int cantidadPrimos = contarPrimos(n, a);
        System.out.println("\nCantidad de números primos:
" + cantidadPrimos);
    }}

```

Corrida:

```

cantidad de elementos: 5
V[0]=1
V[1]=2
V[2]=3
V[3]=4
V[4]=5
|1|2|3|4|5
Cantidad de números primos: 3

```

5. Leer un vector A de n elementos, desplegar los múltiplos de k y las posiciones que ocupan.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer5 { //Leer un vector A de n elementos, desplegar los múltiplos de k y las posiciones que ocupan. public static void mostrarMultiplosDeK(int n, int a[], int k) { System.out.println("Múltiplos de " + k + " y sus posiciones:"); for (int i = 0; i < n; i++) { if (a[i] % k == 0) { System.out.println("a[" + i + "] = " + a[i]); } } } public static void LlenaV(int n,int A[]){ Scanner lee=new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("V["+i+"]"+"="); A[i]=lee.nextInt(); } public static void MostrarV(int m,int b[]){ for (int i = 0; i < m; i++) { System.out.print(" "+b[i]); } } </pre>

```

////////////////////////////////////
////////////////////////////////////
public static void main(String[] args) {
    int n, k;
    Scanner lee = new Scanner(System.in);

    System.out.print("Cantidad de elementos: ");
    n = lee.nextInt();
    int a[] = new int[n];

    LlenaV(n, a);
    MostrarV(n, a);

    System.out.print("\nIngrese el valor de k: ");
    k = lee.nextInt();

    mostrarMultiplosDeK(n, a, k);
}
}

```

Corrida:

```

run:
Cantidad de elementos: 4
V[0]=1
V[1]=2
V[2]=3
V[3]=4
|1|2|3|4
Ingrese el valor de k: 2
Múltiplos de 2 y sus posiciones:
a[1] = 2
a[3] = 4

```

6. Se tiene un vector A de n elementos, eliminar el elemento X que se encuentra en el vector, si el elemento no está en el vector mostrar el mensaje respectivo.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer6 { //Se tiene un vector A de n elementos, eliminar el elemento X que se encuentra en el vector, si el elemento no está en el vector mostrar el mensaje respectivo. public static int eliminarElemento(int n, int a[], int x) { int pos = -1; for (int i = 0; i < n; i++) { if (a[i] == x) { pos = i; break; } } if (pos == -1) { System.out.println("El elemento " + x + " no se encuentra en el vector."); } } } </pre>


```

        return n;
    }

    for (int i = pos; i < n - 1; i++) {
        a[i] = a[i + 1];
    }

    System.out.println("El elemento " + x + " ha sido eliminado.");
    return n - 1; // Devolver el nuevo tamaño del vector
}

public static void LlenaV(int n,int A[]){
    Scanner lee=new Scanner(System.in);

    for (int i = 0; i < n; i++) {
        System.out.print("V["+i+"]"+"=");
        A[i]=lee.nextInt(); }
    public static void MostrarV(int m,int b[]){

        for (int i = 0; i < m; i++) {
            System.out.print("|"+b[i]); }
        //////////////////////////////////////
        //////////////////////////////////////
    }

    public static void main(String[] args) {
        int n, x;
        Scanner lee = new Scanner(System.in);

        System.out.print("Cantidad de elementos: ");
        n = lee.nextInt();
        int a[] = new int[n];

        LlenaV(n, a);
        MostrarV(n, a);

        System.out.print("\nIngresa el elemento a eliminar: ");
        x = lee.nextInt();

        n = eliminarElemento(n, a, x); // Actualizamos el tamaño del vector
        MostrarV(n, a); // Mostrar el vector actualizado
    }
}

```

Corrida:

```

run:
Cantidad de elementos: 5
V[0]=1
V[1]=2
V[2]=3
V[3]=4
V[4]=5
|1|2|3|4|5
Ingresa el elemento a eliminar: 3
El elemento 3 ha sido eliminado.
|1|2|4|5BUILD SUCCESSFUL (total time: 8 seconds)

```

7. Si se tiene un vector A de n elementos, $n > 3$, hallar el promedio cada tres elementos, desplegar el vector.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ </pre>

```
public class Ejer7 {

    //Si se tiene un vector A de n elementos, n>3,
    hallar el promedio cada tres elementos, desplegar el
    vector.
    public static void promedioCadaTres(int n, int a[]) {
        // Asegúrate de que el tamaño del vector sea mayor
        a 3
        if (n < 3) {
            System.out.println("El tamaño del vector debe
            ser mayor que 3.");
            return;
        }

        for (int i = 0; i <= n - 3; i += 3) {
            int suma = a[i] + a[i + 1] + a[i + 2];
            double promedio = suma / 3.0;
            System.out.println("Promedio de los elementos
            " + a[i] + ", " + a[i + 1] + ", " + a[i + 2] + " = " +
            promedio);
        }
    }

    public static void LlenaV(int n,int A[]){
        Scanner lee=new Scanner(System.in);

        for (int i = 0; i < n; i++) {
            System.out.print("V["+i+"]"+"=");
            A[i]=lee.nextInt(); }
        public static void MostrarV(int m,int b[]){

            for (int i = 0; i < m; i++) {
                System.out.print("|"+b[i]); } }

    //////////////////////////////////////
    //////////////////////////////////////
    public static void main(String[] args) {
        int n;
        Scanner lee = new Scanner(System.in);

        System.out.print("Cantidad de elementos: ");
        n = lee.nextInt();
        int a[] = new int[n];

        LlenaV(n, a);
        MostrarV(n, a);

        System.out.println("\nPromedio cada tres
        elementos:");
        promedioCadaTres(n, a);
    }
}
```

Corrida:

```

run:
Cantidad de elementos: 7
V[0]=1
V[1]=2
V[2]=3
V[3]=4
V[4]=5
V[5]=6
V[6]=7
|1|2|3|4|5|6|7
Promedio cada tres elementos:
Promedio de los elementos 1, 2, 3 = 2.0
Promedio de los elementos 4, 5, 6 = 5.0

```

8. Dado un vector A y un número k multiplicar cada elemento de A con el numero k, obtener el resultado en otro vector M.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer8 { //Dado un vector A y un número k multiplicar cada elemento de A con el numero k, obtener el resultado en otro vector M. public static void multiplicarVector(int n, int a[], int k, int m[]) { for (int i = 0; i < n; i++) { m[i] = a[i] * k; } } public static void LlenaV(int n,int A[]){ Scanner lee=new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("V["+i+"]"+"="); A[i]=lee.nextInt(); } public static void MostrarV(int m,int b[]){ for (int i = 0; i < m; i++) { System.out.print(" "+b[i]); } } ////////////////////////////////////// ////////////////////////////////////// public static void main(String[] args) { int n, k; Scanner lee = new Scanner(System.in); System.out.print("Cantidad de elementos: "); n = lee.nextInt(); int a[] = new int[n]; int m[] = new int[n]; LlenaV(n, a); </pre>

```

        MostrarV(n, a);

        System.out.print("\nIngrese el valor de k: ");
        k = lee.nextInt();

        multiplicarVector(n, a, k, m);

        System.out.println("\nVector M (resultado de
multiplicar A por " + k + "):");
        MostrarV(n, m);
    }}

```

Corrida:

```

run:
Cantidad de elementos: 5
V[0]=1
V[1]=2
V[2]=3
V[3]=4
V[4]=5
|1|2|3|4|5
Ingrese el valor de k: 3

Vector M (resultado de multiplicar A por 3):
|3|6|9|12|15BUILD SUCCESSFUL (total time: 8 seconds)

```

9. Llenar dos vectores A y B de N elementos cada uno, sumar el elemento de la primera posición del vector A con el elemento de la primera posición del vector B y así sucesivamente hasta N, almacenar el resultado en un vector C, e imprimir el vector resultante.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ej9 { //Llenar dos vectores A y B de N elementos cada uno, sumar el elemento de la primera posición del vector A con el elemento de la primera posición del vector B y así sucesivamente hasta N, almacenar el resultado en un vector C, e imprimir el vector resultante. public static void sumarVectores(int n, int a[], int b[], int c[]) { for (int i = 0; i < n; i++) { c[i] = a[i] + b[i]; } } public static void LlenaV(int n,int A[]){ Scanner lee=new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("V["+i+"]"+"="); A[i]=lee.nextInt(); }} public static void MostrarV(int m,int b[]){ </pre>

```

        for (int i = 0; i < m; i++) {
            System.out.print("|"+b[i]); } }

////////////////////////////////////
////////////////////////////////////
public static void main(String[] args) {
    int n;
    Scanner lee = new Scanner(System.in);

    System.out.print("Cantidad de elementos: ");
    n = lee.nextInt();
    int a[] = new int[n];
    int b[] = new int[n];
    int c[] = new int[n];

    System.out.println("Llenar el vector A:");
    LlenaV(n, a);
    System.out.println("Llenar el vector B:");
    LlenaV(n, b);

    System.out.println("\nVector A:");
    MostrarV(n, a);
    System.out.println("\nVector B:");
    MostrarV(n, b);

    sumarVectores(n, a, b, c);

    System.out.println("\nVector C (resultado de sumar
A y B):");
    MostrarV(n, c);
}

```

Corrida:

```

run:
Cantidad de elementos: 4
Llenar el vector A:
V[0]=1
V[1]=2
V[2]=3
V[3]=4
Llenar el vector B:
V[0]=1
V[1]=2
V[2]=3
V[3]=4

Vector A:
|1|2|3|4
Vector B:
|1|2|3|4
Vector C (resultado de sumar A y B):
|2|4|6|8BUILD SUCCESSFUL (total time: 9 seconds)

```

10. Llenar dos vectores A y B de N elementos cada uno, sumar el elemento de la primera posición del vector A con el elemento de la primera posición del vector B y así sucesivamente hasta N, almacenar el resultado en un vector C, e imprimir el vector resultante.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer10 { //Llenar dos vectores A y B de N elementos cada uno, sumar el elemento de la primera posición del vector A con el elemento de la primera posición del vector B y así sucesivamente hasta N, almacenar el resultado en un vector C, e imprimir el vector resultante. public static void LlenaV(int n, int a[]) { Scanner lee = new Scanner(System.in); for (int i = 0; i < n; i++) { System.out.print("Ingrese el elemento en la posición " + i + ": "); a[i] = lee.nextInt(); } } public static void MostrarV(int n, int a[]) { System.out.print("["); for (int i = 0; i < n; i++) { System.out.print(a[i]); if (i < n - 1) { System.out.print(", "); } } System.out.println("]"); } public static void sumarVectores(int n, int a[], int b[], int c[]) { for (int i = 0; i < n; i++) { c[i] = a[i] + b[i]; } } ////////////////////////////////////// ////////////////////////////////////// public static void main(String[] args) { Scanner lee = new Scanner(System.in); System.out.print("Cantidad de elementos en los vectores: "); int n = lee.nextInt(); int a[] = new int[n]; int b[] = new int[n]; </pre>

```

int c[] = new int[n];

System.out.println("Llenar el vector A:");
LlenaV(n, a);

System.out.println("Llenar el vector B:");
LlenaV(n, b);

sumarVectores(n, a, b, c);

System.out.println("\nVector A:");
MostrarV(n, a);
System.out.println("Vector B:");
MostrarV(n, b);
System.out.println("Vector C (resultado de
sumar A y B):");
MostrarV(n, c);
    }
}

```

Corrida:

```

run:
Cantidad de elementos en los vectores: 4
Llenar el vector A:
Ingrese el elemento en la posición 0: 1
Ingrese el elemento en la posición 1: 2
Ingrese el elemento en la posición 2: 3
Ingrese el elemento en la posición 3: 4
Llenar el vector B:
Ingrese el elemento en la posición 0: 1
Ingrese el elemento en la posición 1: 2
Ingrese el elemento en la posición 2: 3
Ingrese el elemento en la posición 3: 4

Vector A:
[1, 2, 3, 4]
Vector B:
[1, 2, 3, 4]
Vector C (resultado de sumar A y B):
[2, 4, 6, 8]
BUILD SUCCESSFUL (total time: 12 seconds)

```

11. En un vector de números enteros queremos poner al principio todos los números impares y al final los números pares conservando ambos su orden relativo en el vector original. Diseñar una función que obtenga el vector según se ha indicado anteriormente.

Análisis y diseño	Programa
	<pre> public class Ejer11 { public static int[] reordenarVector(int n, int a[]) { int numImpares = 0; int i = 0; while (i < n) { if (a[i] % 2 != 0) { numImpares++; } } } } </pre>

```

        }
        i++;
    }

    int[] impares = new int[numImpares];
    int[] pares = new int[n - numImpares];

    int indexImpares = 0;
    int indexPares = 0;
    i = 0;
    while (i < n) {
        if (a[i] % 2 != 0) {
            impares[indexImpares++] = a[i];
        } else {
            pares[indexPares++] = a[i];
        }
        i++;
    }

    int[] resultado = new int[n];
    int indexResultado = 0;
    i = 0;
    while (i < numImpares) {
        resultado[indexResultado++] =
impares[i++];
    }
    i = 0;
    while (i < n - numImpares) {
        resultado[indexResultado++] = pares[i++];
    }

    return resultado;
}

public static void MostrarV(int n, int a[]) {
    int i = 0;
    System.out.print("[");
    while (i < n) {
        System.out.print(a[i]);
        if (i < n - 1) {
            System.out.print(", ");
        }
        i++;
    }
    System.out.println("]");
}

public static void main(String[] args) {
    int n;
    Scanner lee = new Scanner(System.in);

    System.out.print("Cantidad de elementos: ");
    n = lee.nextInt();
    int[] a = new int[n];

    int i = 0;
    while (i < n) {
        System.out.print("Elemento " + i + ": ");
        a[i] = lee.nextInt();
        i++;
    }

    int[] resultado = reordenarVector(n, a);

    System.out.println("\nVector original:");
    MostrarV(n, a);
    System.out.println("Vector reordenado (impares
al principio, pares al final:");
    MostrarV(n, resultado);
}
}

```

Corrida:


```

run:
Cantidad de elementos: 3
Llenar el vector:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 3

Vector original:
[1, 2, 3]
Vector reordenado (impares al principio, pares al final):
[1, 3, 2]

```

12. Llenar un vector de N elementos, imprimir la posición y el valor del elemento mayor almacenado en el vector. Suponga que todos los elementos del vector son diferentes.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer12 { /** * Llenar un vector de N elementos, imprimir la posición y el valor del * elemento mayor almacenado en el vector. Suponga que todos los elementos * del vector son diferentes. */ public static void encontrarMayor(int n, int a[]) { if (n <= 0) { System.out.println("El tamaño del vector debe ser mayor que 0."); return; } int mayor = a[0]; int posicion = 0; for (int i = 1; i < n; i++) { if (a[i] > mayor) { mayor = a[i]; posicion = i; } } System.out.println("El valor del elemento mayor es: " + mayor); System.out.println("La posición del elemento mayor es: " + posicion); } public static void MostrarV(int n, int a[]) { System.out.print("["); for (int i = 0; i < n; i++) { System.out.print(a[i]); if (i < n - 1) { System.out.print(", "); } } System.out.println("]"); } public static void main(String[] args) { </pre>

```

int n;
Scanner lee = new Scanner(System.in);

System.out.print("Cantidad de elementos: ");
n = lee.nextInt();
int a[] = new int[n];

System.out.println("Llenar el vector:");
for (int i = 0; i < n; i++) {
    System.out.print("Elemento " + i + ": ");
    a[i] = lee.nextInt();
}

System.out.println("\nVector:");
MostrarV(n, a);

encontrarMayor(n, a);
}
}

```

Corrida:

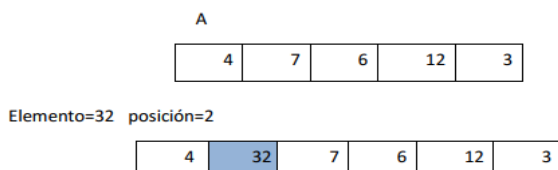
```

run:
Cantidad de elementos: 5
Llenar el vector:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 3
Elemento 3: 4
Elemento 4: 5

Vector:
[1, 2, 3, 4, 5]
El valor del elemento mayor es: 5
La posición del elemento mayor es: 4

```

13. Se tiene un vector A de N elementos, se pide dos números que será el elemento y la posición donde será insertado en el vector, si la posición no existe añadir el elemento al final.



Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer13 { /** * Se tiene un vector A de N elementos, se pide dos números que será el </pre>

```
* elemento y la posición donde será
insertado en el vector, si la posición no existe
añadir el elemento al final.

*/
public static void insertarElemento(int n, int[]
a, int elemento, int posicion) {

    int[] nuevoVector = new int[n + 1];

    if (posicion < 0) {
        posicion = 0;
    } else if (posicion >= n) {
        posicion = n;
    }

    for (int i = 0; i < posicion; i++) {
        nuevoVector[i] = a[i];
    }

    nuevoVector[posicion] = elemento;

    for (int i = posicion; i < n; i++) {
        nuevoVector[i + 1] = a[i];
    }

    System.out.println("Nuevo vector:");
    MostrarV(n + 1, nuevoVector);
}

public static void MostrarV(int n, int[] a) {
    System.out.print("[");
    for (int i = 0; i < n; i++) {
        System.out.print(a[i]);
        if (i < n - 1) {
            System.out.print(", ");
        }
    }
    System.out.println("]");
}

public static void main(String[] args) {
    int n;
    Scanner lee = new Scanner(System.in);

    System.out.print("Cantidad de elementos en el
vector: ");
    n = lee.nextInt();
    int a[] = new int[n];

    System.out.println("Llenar el vector:");
    for (int i = 0; i < n; i++) {
        System.out.print("Elemento " + i + ": ");
        a[i] = lee.nextInt();
    }
}
```

```

        System.out.print("Ingrese el nuevo elemento:
");
        int nuevoElemento = lee.nextInt();

        System.out.print("Ingrese la posición donde
insertar el nuevo elemento: ");
        int posicion = lee.nextInt();

        insertarElemento(n, a, nuevoElemento,
posicion);
    }
}

```

Corrida:

```

run:
Cantidad de elementos en el vector: 5
Llenar el vector:
Elemento 0: 4
Elemento 1: 7
Elemento 2: 6
Elemento 3: 12
Elemento 4: 3
Ingrese el nuevo elemento: 32
Ingrese la posición donde insertar el nuevo elemento: 2
Nuevo vector:
[4, 7, 32, 6, 12, 3]
BUILD SUCCESSFUL (total time: 22 seconds)

```

14. En un vector de eliminar los elementos repetidos guardando el resultado en el mismo vector. Los elementos deben conservar su orden relativo. Para ello no se puede utilizar ningún vector auxiliar.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ej14 { /** * @En un vector de eliminar los elementos repetidos guardando el resultado * en el mismo vector. Los elementos deben conservar su orden relativo. * Para ello no se puede utilizar ningún vector auxiliar. */ public static void eliminarRepetidos(int n, int[] a) { int i, j; int nuevoTamano = 0; </pre>

```
for (i = 0; i < n; i++) {
    boolean repetido = false;
    for (j = 0; j < nuevoTamano; j++) {
        if (a[i] == a[j]) {
            repetido = true;
            break;
        }
    }
    if (!repetido) {
        a[nuevoTamano++] = a[i];
    }
    for (i = nuevoTamano; i < n; i++) {
        a[i] = 0;
    }

    System.out.println("Vector sin elementos
repetidos:");
    MostrarV(nuevoTamano, a);
}

public static void MostrarV(int n, int[] a) {
    System.out.print("[");
    for (int i = 0; i < n; i++) {
        System.out.print(a[i]);
        if (i < n - 1) {
            System.out.print(", ");
        }
    }
    System.out.println("]");
}

public static void main(String[] args) {
    int n;
    Scanner lee = new Scanner(System.in);

    System.out.print("Cantidad de elementos en el
vector: ");
    n = lee.nextInt();
    int a[] = new int[n];

    System.out.println("Llenar el vector:");
    for (int i = 0; i < n; i++) {
        System.out.print("Elemento " + i + ": ");
        a[i] = lee.nextInt();
    }

    eliminarRepetidos(n, a);
}
```

Corrida:

```

run:
Cantidad de elementos en el vector: 4
Llenar el vector:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 2
Elemento 3: 3
Vector sin elementos repetidos:
[1, 2, 3]
BUILD SUCCESSFUL (total time: 4 seconds)

```

15. Escribir un programa que lea un vector de 10 elementos. Deberá imprimir el mismo vector por pantalla pero invertido. Ejemplo: dado el vector 1 2 3 4 5 6 7 8 9 10 el programa debería imprimir 10 9 8 7 6 5 4 3 2 1. (no utilizar un vector auxiliar).

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer15 { /** * . Escribir un programa que lea un vector de 10 elementos. * Deberá imprimir el mismo vector por pantalla pero invertido. Ejemplo: dado el vector 1 2 3 4 5 6 7 8 9 10 el programa * debería imprimir 10 9 8 7 6 5 4 3 2 1. (no utilizar un vector auxiliar). */ public static void invertirYMostrar(int n, int[] a) { int inicio = 0; int fin = n - 1; while (inicio < fin) { int temp = a[inicio]; a[inicio] = a[fin]; a[fin] = temp; inicio++; fin--; } System.out.println("Vector invertido:"); MostrarV(n, a); } public static void MostrarV(int n, int[] a) { System.out.print("["); </pre>

```

        for (int i = 0; i < n; i++) {
            System.out.print(a[i]);
            if (i < n - 1) {
                System.out.print(", ");
            }
            System.out.println("");
        }

        public static void main(String[] args) {
            int n = 10;
            int[] a = new int[n];
            Scanner lee = new Scanner(System.in);

            System.out.println("Llenar el vector de 10
elementos:");
            for (int i = 0; i < n; i++) {
                System.out.print("Elemento " + i + ": ");
                a[i] = lee.nextInt();
            }

            invertirYMostrar(n, a);
        }
    }
}

```

Corrida:

```

run:
Llenar el vector de 10 elementos:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 3
Elemento 3: 4
Elemento 4: 5
Elemento 5: 6
Elemento 6: 7
Elemento 7: 8
Elemento 8: 9
Elemento 9: 10
Vector invertido:
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
BUILD SUCCESSFUL (total time: 8 seconds)

```

16. Dados dos vectores A de n elementos y un vector B de n elementos, determinar cuántas veces la diferencia entre elementos iguales de los dos vectores es menor, igual o mayor a cero. Desplegar resultados.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer16 { /** </pre>

```
* Dados dos vectores A de n elementos y un vector
B de n elementos,
* determinar cuántas veces la diferencia entre
elementos iguales de los
* dos vectores es menor, igual o mayor a cero.
Desplegar resultados.
*/
public static void calcularDiferencias(int n,
int[] a, int[] b) {
    int menorCero = 0;
    int igualCero = 0;
    int mayorCero = 0;

    for (int i = 0; i < n; i++) {
        int diferencia = a[i] - b[i];
        if (diferencia < 0) {
            menorCero++;
        } else if (diferencia == 0) {
            igualCero++;
        } else {
            mayorCero++;
        }
    }

    System.out.println("Cantidad de diferencias
menores a cero: " + menorCero);
    System.out.println("Cantidad de diferencias
iguales a cero: " + igualCero);
    System.out.println("Cantidad de diferencias
mayores a cero: " + mayorCero);
}

public static void MostrarV(int n, int[] a) {
    System.out.print("[");
    for (int i = 0; i < n; i++) {
        System.out.print(a[i]);
        if (i < n - 1) {
            System.out.print(", ");
        }
    }
    System.out.println("]");
}

public static void main(String[] args) {
    int n;
    Scanner lee = new Scanner(System.in);

    System.out.print("Cantidad de elementos en los
vectores: ");
    n = lee.nextInt();

    int[] a = new int[n];
    int[] b = new int[n];
    System.out.println("Llenar el vector A:");
    for (int i = 0; i < n; i++) {
        System.out.print("Elemento " + i + " de A:
");
        a[i] = lee.nextInt();
    }

    System.out.println("Llenar el vector B:");
    for (int i = 0; i < n; i++) {
```



```

        System.out.print("Elemento " + i + " de B:
");
        b[i] = lee.nextInt();
    }

    System.out.println("\nVector A:");
    MostrarV(n, a);
    System.out.println("Vector B:");
    MostrarV(n, b);

    calcularDiferencias(n, a, b);
}
}

```

Corrida:

```

run:
Cantidad de elementos en los vectores: 3
Llenar el vector A:
Elemento 0 de A: 1
Elemento 1 de A: 2
Elemento 2 de A: 3
Llenar el vector B:
Elemento 0 de B: 1
Elemento 1 de B: 2
Elemento 2 de B: 3

Vector A:
[1, 2, 3]
Vector B:
[1, 2, 3]
Cantidad de diferencias menores a cero: 0
Cantidad de diferencias iguales a cero: 3
Cantidad de diferencias mayores a cero: 0
BUILD SUCCESSFUL (total time: 6 seconds)

```

17. Leer N pares de números X y Z, el mayor de X y Z almacenar como elemento de un vector A y el menor almacenar en el vector B. Desplegar los vectores.

Análisis y diseño	Programa
	<pre> public static void mostrarVector(int[] a, int n) { int i = 0; System.out.print("["); while (i < n) { System.out.print(a[i]); if (i < n - 1) { System.out.print(", "); } i++; } System.out.println("]"); } public static void main(String[] args) { int n; Scanner scanner = new Scanner(System.in); </pre>

```

        System.out.print("Cantidad de pares de
números: ");
        n = scanner.nextInt();

        int[] A = new int[n];
        int[] B = new int[n];

        int i = 0;
        while (i < n) {
            System.out.print("Ingrese el número X para
el par " + (i + 1) + ": ");
            int X = scanner.nextInt();
            System.out.print("Ingrese el número Z para
el par " + (i + 1) + ": ");
            int Z = scanner.nextInt();
            if (X > Z) {
                A[i] = X;
                B[i] = Z;
            } else {
                A[i] = Z;
                B[i] = X;
            }
            i++;
        }
        System.out.println("Vector A (mayores):");
        mostrarVector(A, n);
        System.out.println("Vector B (menores):");
        mostrarVector(B, n);
    }
}

```

Corrida:

```

run:
Cantidad de pares de números: 3
Ingrese el número X para el par 1: 1
Ingrese el número Z para el par 1: 2
Ingrese el número X para el par 2: 3
Ingrese el número Z para el par 2: 1
Ingrese el número X para el par 3: 2
Ingrese el número Z para el par 3: 3
Vector A (mayores):
[2, 3, 3]
Vector B (menores):
[1, 1, 2]
BUILD SUCCESSFUL (total time: 16 seconds)

```

18. Almacenar N números en un vector, imprimir cuantos son ceros, cuántos son negativos, cuantos positivos.
Imprimir además la suma de los negativos y la suma de los positivos.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; </pre>

```
/**
 *
 * @author DELL
 */
public class Ejer18 {

    /**
     * Almacenar N números en un vector, imprimir
     * cuantos son ceros,
     * cuántos son negativos, cuantos positivos.
     * Imprimir además la suma de los negativos y la suma de
     * los positivos.
     */

    public static void main(String[] args) {
        int n;
        Scanner scanner = new Scanner(System.in);

        System.out.print("Cantidad de números en el
vector: ");
        n = scanner.nextInt();

        int[] vector = new int[n];
        int contadorCeros = 0;
        int contadorNegativos = 0;
        int contadorPositivos = 0;
        int sumaNegativos = 0;
        int sumaPositivos = 0;

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

            if (vector[i] == 0) {
                contadorCeros++;
            } else if (vector[i] < 0) {
                contadorNegativos++;
                sumaNegativos += vector[i];
            } else {
                contadorPositivos++;
                sumaPositivos += vector[i];
            }
        }

        System.out.println("Cantidad de ceros: " +
contadorCeros);
        System.out.println("Cantidad de negativos: " +
contadorNegativos);
        System.out.println("Cantidad de positivos: " +
contadorPositivos);
        System.out.println("Suma de los negativos: " +
sumaNegativos);
        System.out.println("Suma de los positivos: " +
sumaPositivos);
    }
}
```

Corrida:

```

run:
Cantidad de números en el vector: 5
Ingrese los números:
Número 1: 1
Número 2: 2
Número 3: 3
Número 4: 4
Número 5: 5
Cantidad de ceros: 0
Cantidad de negativos: 0
Cantidad de positivos: 5
Suma de los negativos: 0
Suma de los positivos: 15
BUILD SUCCESSFUL (total time: 6 seconds)

```

19.

1. Se tienen dos vectores A y B de N elementos cada uno. Hacer un algoritmo que escriba la palabra "iguales" si ambos vectores son iguales y "Diferentes" si no lo son. Serán iguales cuando en la misma posición de ambos vectores se tenga el mismo valor para todos los elementos.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ej19 { /** * Se tienen dos vectores A y B de N elementos * cada uno. * Hacer un algoritmo que escriba la palabra * "iguales" si ambos vectores son iguales y "Diferentes" * si no lo son. Serán iguales cuando en la misma * posición de ambos vectores se tenga el mismo valor * para todos los elementos */ public static void compararVectores(int n, int[] a, int[] b) { boolean iguales = true; for (int i = 0; i < n; i++) { if (a[i] != b[i]) { iguales = false; break; } } if (iguales) { System.out.println("iguales"); } else { </pre>

```

        System.out.println("diferentes");
    }
}

public static void main(String[] args) {
    int n;
    Scanner scanner = new Scanner(System.in);

    System.out.print("Cantidad de elementos en los
vectores: ");
    n = scanner.nextInt();

    int[] a = new int[n];
    int[] b = new int[n];

    System.out.println("Llenar el vector A:");
    for (int i = 0; i < n; i++) {
        System.out.print("Elemento " + i + " de A:
");
        a[i] = scanner.nextInt();
    }

    System.out.println("Llenar el vector
B:");
    for (int i = 0; i < n; i++) {
        System.out.print("Elemento " + i + " de B:
");
        b[i] = scanner.nextInt();
    }

    compararVectores(n, a, b);
}
}

```

Corrida:

```

run:
Cantidad de elementos en los vectores: 3
Llenar el vector A:
Elemento 0 de A: 1
Elemento 1 de A: 2
Elemento 2 de A: 3
Llenar el vector B:
Elemento 0 de B: 1
Elemento 1 de B: 2
Elemento 2 de B: 3
iguales
BUILD SUCCESSFUL (total time: 5 seconds)

```

20.

Lea un número cualquiera y lo busque en el vector V, el cual tiene almacenados N elementos.

Escribir la posición donde se encuentra almacenado el número en el vector o el mensaje "NO" si no lo encuentra.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ </pre>

```
public class Ejer20 {

    /**
     * Lea un número cualquiera y lo busque en el
     * vector V, el cual tiene
     * almacenados N elementos. Escribir la posición
     * donde se encuentra
     * almacenado el número en el vector o el mensaje
     * "NO" si no lo encuentra

     */
    public static void buscarNumero(int n, int[]
vector, int numeroABuscar) {
        boolean encontrado = false;
        for (int i = 0; i < n; i++) {
            if (vector[i] == numeroABuscar) {
                System.out.println("El número se
encuentra en la posición: " + i);
                encontrado = true;
                break;
            }
        }

        if (!encontrado) {
            System.out.println("NO");
        }
    }

    public static void main(String[] args) {
        int n;
        Scanner scanner = new Scanner(System.in);

        System.out.print("Cantidad de elementos
en el vector: ");
        n = scanner.nextInt();

        int[] vector = new int[n];

        System.out.println("Ingrese los elementos del
vector:");
        for (int i = 0; i < n; i++) {
            System.out.print("Elemento " + i + ": ");
            vector[i] = scanner.nextInt();
        }

        System.out.print("Ingrese el número a buscar:
");
        int numeroABuscar = scanner.nextInt();

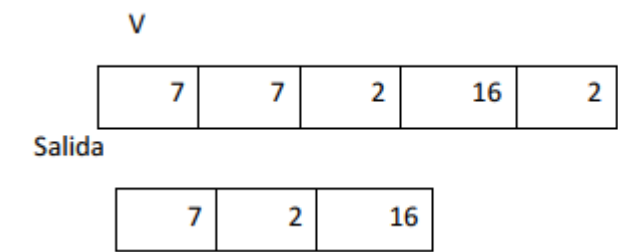
        buscarNumero(n, vector, numeroABuscar);
    }
}
```

Corrida:

```
run:
Cantidad de elementos en el vector: 3
Ingrese los elementos del vector:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 3
Ingrese el número a buscar: 3
El número se encuentra en la posición: 2
BUILD SUCCESSFUL (total time: 6 seconds)
```

21.

Se tiene un vector V con N elementos, se pide eliminar los elementos repetidos del vector, mostrar el vector resultante.



Análisis y diseño	Programa
	<pre>package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ej21 { /** * @param args the command line arguments */ public static void eliminarRepetidos(int[] vector, int n) { int index = 0; for (int i = 0; i < n; i++) { boolean repetido = false; for (int j = 0; j < index; j++) { if (vector[i] == vector[j]) { repetido = true; break; } } if (!repetido) { vector[index++] = vector[i]; } } System.out.println("Vector resultante sin elementos repetidos:"); for (int i = 0; i < index; i++) { System.out.print(vector[i] + " "); } System.out.println(); } }</pre>

```

    }

    public static void main(String[] args) {
        int n;
        Scanner scanner = new Scanner(System.in);

        System.out.print("Cantidad de elementos en el
vector: ");
        n = scanner.nextInt();

        int[] vector = new int[n];

        System.out.println("Ingrese los elementos del
vector:");
        for (int i = 0; i < n; i++) {
            System.out.print("Elemento " + i + ": ");
            vector[i] = scanner.nextInt();
        }

        eliminarRepetidos(vector, n);
    }
}

```

Corrida:

```

run:
Cantidad de elementos en el vector: 4
Ingrese los elementos del vector:
Elemento 0: 1
Elemento 1: 21
Elemento 2: 2
Elemento 3: 1
Vector resultante sin elementos repetidos:
1 21 2
BUILD SUCCESSFUL (total time: 10 seconds)

```

22*. Realizar la suma de dos vectores considerando que son de distinta longitud.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer22 { /** * Realizar la suma de dos vectores considerando que son de distinta * longitud */ public static void sumarVectores(int[] vectorA, int[] vectorB, int nA, int nB) { int MAX_SIZE = 100; int[] vectorResultado = new int[MAX_SIZE]; </pre>


```
for (int i = 0; i < MAX_SIZE; i++) {
    vectorResultado[i] = 0;
}

for (int i = 0; i < nA; i++) {
    vectorResultado[i] += vectorA[i];
}
for (int i = 0; i < nB; i++) {
    vectorResultado[i] += vectorB[i];
}

    System.out.println("Vector resultante
de la suma:");
    for (int i = 0; i < (nA > nB ? nA : nB); i++)
    {
        System.out.print(vectorResultado[i] + "
");
    }
    System.out.println();
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Cantidad de
elementos en el primer vector: ");
    int nA = scanner.nextInt();
    int[] vectorA = new int[nA];

    System.out.println("Ingrese los elementos del
primer vector:");
    for (int i = 0; i < nA; i++) {
        System.out.print("Elemento " + i + ": ");
        vectorA[i] = scanner.nextInt();
    }
    System.out.print("Cantidad de elementos en el
segundo vector: ");
    int nB = scanner.nextInt();
    int[] vectorB = new int[nB];

    System.out.println("Ingrese los elementos del
segundo vector:");
    for (int i = 0; i < nB; i++) {
        System.out.print("Elemento " + i + ": ");
        vectorB[i] = scanner.nextInt();
    }

    sumarVectores(vectorA, vectorB, nA,
nB);
}
```

Corrida:

```

run:
Cantidad de elementos en el primer vector: 3
Ingrese los elementos del primer vector:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 3
Cantidad de elementos en el segundo vector: 4
Ingrese los elementos del segundo vector:
Elemento 0: 12
Elemento 1: 2
Elemento 2: 3
Elemento 3: 4
Vector resultante de la suma:
13 4 6 4
BUILD SUCCESSFUL (total time: 11 seconds)

```

23. Generar y desplegar un vector de N elementos que contenga los números de la serie de Fibonacci.

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * * @author DELL */ public class Ejer23 { /** * Generar y desplegar un vector de N elementos que contenga los * números de la serie de Fibonacci. */ public static void llenarFibonacci(int[] vector, int n) { if (n > 0) { vector[0] = 0; } if (n > 1) { vector[1] = 1; } for (int i = 2; i < n; i++) { vector[i] = vector[i - 1] + vector[i - 2]; } } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Cantidad de elementos Fibonacci: "); int n = scanner.nextInt(); int[] vectorFibonacci = new int[n]; </pre>

```

        llenarFibonacci(vectorFibonacci, n);

        System.out.println("Vector de Fibonacci:");
        for (int i = 0; i < n; i++) {
            System.out.print(vectorFibonacci[i] + "
");
        }
        System.out.println();
    }
}

```

Corrida:

```

run:
Cantidad de elementos Fibonacci: 5
Vector de Fibonacci:
0 1 1 2 3
BUILD SUCCESSFUL (total time: 1 second)

```

24. ***

Análisis y diseño	Programa

Corrida:

```

run:
Cantidad de elementos: 5
Vector de números primos:
2 3 5 7 11

```

25. ***

Generar y desplegar un vector de N elementos que tenga la forma siguiente

1	4	1	16	1	36	1	64	1	100
---	---	---	----	---	----	---	----	---	-----

Análisis y diseño	Programa
	<pre> package javaapplication25; import java.util.Scanner; /** * Generar y desplegar un vector de N elementos que * tenga la forma siguiente: * 1 4 1 16 1 36 1 64 1 100 */ public class Ej25 { /** * Llenar el vector con el patrón especificado. */ public static void llenarVector(int[] vector, int n) { </pre>

```

int num = 2;
int i = 0;

while (i < n) {
    if (i % 2 == 0) {
        vector[i] = 1;
    } else {
        vector[i] = num * num;
        num += 2;
    }
    i++;
}

public static void mostrarVector(int[] vector, int
n) {
    System.out.println("Vector generado:");
    int i = 0;
    while (i < n) {
        System.out.print(vector[i] + " ");
        i++;
    }
    System.out.println();
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Cantidad de elementos en el
vector: ");
    int n = scanner.nextInt();

    int[] vector = new int[n];

    llenarVector(vector, n);

    mostrarVector(vector, n);
}

```

Corrida:

```

run:
Cantidad de elementos en el vector: 10
Vector generado:
1 4 1 16 1 36 1 64 1 100
BUILD SUCCESSFUL (total time: 2 seconds)

```

26. ***

Generar y desplegar un vector de N elementos que tenga la forma siguiente

0	1	0	1	1	0	1	1	1	0
---	---	---	---	---	---	---	---	---	---

Análisis y diseño	Programa

Corrida:

```

run:
Cantidad de elementos en el vector: 10
Vector generado:
0 1 0 1 1 0 1 1 1 0
BUILD SUCCESSFUL (total time: 7 seconds)

```

27.*** Generar y desplegar un vector de N elementos que tenga la forma siguiente

1	4	7	10	13	16	19	21	24	27
---	---	---	----	----	----	----	----	----	----

Análisis y diseño	Programa

Corrida:

```
run:
Cantidad de elementos en el vector: 10
Vector generado:
1 4 7 10 13 16 19 22 24 26
BUILD SUCCESSFUL (total time: 1 second)
```

28.

Generar y desplegar un vector de N elementos que tenga la forma siguiente

2	5	3	4	10	6	6	15	9	8
---	---	---	---	----	---	---	----	---	---

Análisis y diseño	Programa

Corrida:

29.

Generar y desplegar un vector de N elementos que tenga la forma siguiente. Dado un valor de inicio. Por ejemplo 8 convertir a base 5. Generar los números consecutivos a dicha base.

13	14	20	21	22	23	24	30	31	32
----	----	----	----	----	----	----	----	----	----

Análisis y diseño	Programa

Corrida:

30.

Generar y desplegar un vector de N elementos que tenga la forma siguiente

5	3	1	6	4	2	7	5	3	8
---	---	---	---	---	---	---	---	---	---

Análisis y diseño	Programa

Corrida: