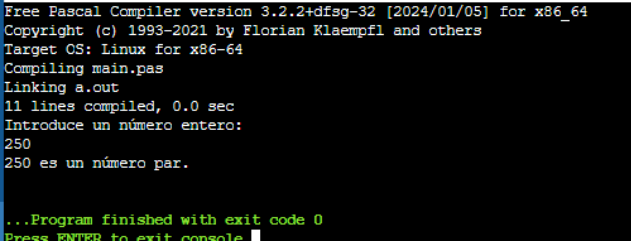
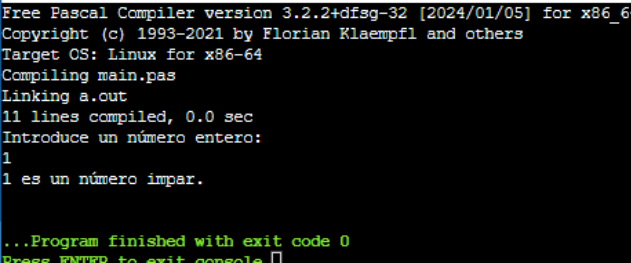


Reporte de Examen práctico.

Problema #1: Determinar si un número es par o impar. (5 puntos)

Nombre del alumno(a):	Caleb Leonardo Pitalúa Salazar	Fecha:	05/09/2025
-----------------------	--------------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre> program ParImpar ! Caleb pitalua implicit none integer :: numero write(*,*) 'Ingrese un número:' read(*,*) numero if (mod(numero, 2) == 0) then write(*,*) numero, ' es un número par.' else write(*,*) numero, ' es un número impar.' end if end program ParImpar </pre>	<div> <p>STDIN</p> <p>1</p> <hr/> <p>Output:</p> <p>Ingrese un número: 1 es un número impar.</p> </div> <div> <p>STDIN</p> <p>250</p> <hr/> <p>Output:</p> <p>Ingrese un número: 250 es un número par.</p> </div>

Código en el lenguaje Pascal	Ejecución
<pre>{Caleb Leonardo Pitalúa Salazar} PROGRAM ParImparPascal; VAR numero: INTEGER; BEGIN WriteLn('Introduce un número entero:'); ReadLn(numero); IF (numero MOD 2 = 0) THEN WriteLn(numero, ' es un número par.') ELSE WriteLn(numero, ' es un número impar.');</pre> <p>END.</p>	 

Código en el lenguaje C/C++	Ejecución
-----------------------------	-----------

<pre>#include <stdio.h> int main() { int numero; printf("Ingrese un número: "); scanf("%d", &numero); if (numero % 2 == 0) { printf("%d es un número par.\n", numero); } else { printf("%d es un número impar.\n", numero); } return 0; }</pre>	<div>2</div> <div>Salida del programa</div> <div>Ingrese un número: 2 es un número par.</div> <div>[Execution complete with exit code 0]</div> <div>999</div> <div>Salida del programa</div> <div>Ingrese un número: 999 es un número impar.</div> <div>[Execution complete with exit code 0]</div>
--	---

Código en el lenguaje Java	Ejecución
----------------------------	-----------

```
import java.util.Scanner;
//Caleb Pitalua
public class ParImparJava {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Introduce un número
entero:");
        int numeros = scanner.nextInt();

        if (numeros % 2 == 0) {
            System.out.println(numeros + " es un
número par.");
        } else {
            System.out.println(numeros + " es un
número impar.");
        }

        scanner.close();
    }
}
```

```
Introduce un n?mero entero:
1000
1000 es un n?mero par.

=== Code Execution Successful ===
```

Output

```
Introduce un n?mero entero:
555
555 es un n?mero impar.

=== Code Execution Successful ===
```

Reporte de Examen práctico.

Problema # 2: Convertir grados Celsius a Fahrenheit. (5 puntos)

Nombre del alumno(a):	Caleb Leonardo Pitalúa Salazar	Fecha:	05/09/2025
-----------------------	--------------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre>PROGRAM CELSIUS_A_FAHRENHEIT !Caleb Leonardo Pitalúa Salazar IMPLICIT NONE REAL :: celsius, fahrenheit ! Solicitar al usuario que ingrese la temperatura en Celsius WRITE(*,*) 'Ingrese la temperatura en grados Celsius:' READ(*,*) celsius ! Calcular la temperatura en Fahrenheit fahrenheit = (celsius * 9.0 / 5.0) + 32.0 ! Mostrar el resultado WRITE(*,*) 'La temperatura en Fahrenheit es:', fahrenheit END PROGRAM CELSIUS_A_FAHRENHEIT</pre>	<p>50</p> <hr/> <p>Output:</p> <p>Ingrese la temperatura en grados Celsius: La temperatura en Fahrenheit es: 122.000000</p> <hr/> <p>0</p> <hr/> <p>Output:</p> <p>Ingrese la temperatura en grados Celsius: La temperatura en Fahrenheit es: 32.000000</p>

Código en el lenguaje Pascal	Ejecución
<pre>{Caleb Leonardo Pitalúa Salazar} PROGRAM CelsiusToFahrenheit; VAR celsius, fahrenheit: Real; BEGIN // Solicitar al usuario que ingrese la temperatura en Celsius WriteLn('Ingrese la temperatura en grados Celsius:'); ReadLn(celsius); // Calcular la temperatura en Fahrenheit fahrenheit := (celsius * 9.0 / 5.0) + 32.0; // Mostrar el resultado WriteLn('La temperatura en Fahrenheit es: ', fahrenheit:0:2); // :0:2 para dos decimales END.</pre>	<pre>Free Pascal Compiler version 3.2.2+dfsg-32 [2024/01/05] for x86_64 Copyright (c) 1993-2021 by Florian Klaempfl and others Target OS: Linux for x86-64 Compiling main.pas Linking a.out 13 lines compiled, 0.0 sec Ingrese la temperatura en grados Celsius: 50 La temperatura en Fahrenheit es: 122.00 ...Program finished with exit code 0 Press ENTER to exit console.</pre> <pre>Free Pascal Compiler version 3.2.2+dfsg-32 [2024/01/05] Copyright (c) 1993-2021 by Florian Klaempfl and others Target OS: Linux for x86-64 Compiling main.pas Linking a.out 13 lines compiled, 0.0 sec Ingrese la temperatura en grados Celsius: -5 La temperatura en Fahrenheit es: 23.00 ...Program finished with exit code 0 Press ENTER to exit console.█</pre>

Código en el lenguaje C/C++	Ejecución
<pre>#include <stdio.h> int main() { float celsius, fahrenheit; // Solicitar al usuario que ingrese la temperatura en Celsius printf("Ingrese la temperatura en grados Celsius: "); scanf("%f", &celsius); // Calcular la temperatura en Fahrenheit fahrenheit = (celsius * 9.0 / 5.0) + 32.0; // Mostrar el resultado printf("La temperatura en Fahrenheit es: %.2f\n", fahrenheit); // .2f para dos decimales return 0; }</pre>	<p>0</p> <hr/> <p>Salida del programa</p> <div> <p>Ingrese la temperatura en grados Celsius: La temperatura en Fahrenheit es: 32.00</p> <p>[Execution complete with exit code 0]</p> </div> <hr/> <p>-55</p> <hr/> <p>Salida del programa</p> <div> <p>Ingrese la temperatura en grados Celsius: La temperatura en Fahrenheit es: -67.00</p> <p>[Execution complete with exit code 0]</p> </div>

Código en el lenguaje Java	Ejecución
<pre>import java.util.Scanner; public class CelsiusToFahrenheit { public static void main(String[] args) { Scanner input = new Scanner(System.in); double celsius, fahrenheit; // Solicitar al usuario que ingrese la temperatura en Celsius System.out.print("Ingrese la temperatura en grados Celsius: "); celsius = input.nextDouble(); // Calcular la temperatura en Fahrenheit fahrenheit = (celsius * 9.0 / 5.0) + 32.0; // Mostrar el resultado System.out.printf("La temperatura en Fahrenheit es: %.2f%n", fahrenheit); // .2f para dos decimales input.close(); } }</pre>	<div> <p>Output</p> <pre>Ingrese la temperatura en grados Celsius: -250 La temperatura en Fahrenheit es: -418.00 === Code Execution Successful ===</pre> </div> <div> <p>Output</p> <pre>Ingrese la temperatura en grados Celsius: 750 La temperatura en Fahrenheit es: 1382.00 === Code Execution Successful ===</pre> </div>

Reporte de Examen práctico.

Problema 4.- Mostrar los primeros n términos de la serie de Fibonacci. (10 puntos)

Nombre del alumno(a):	Caleb Leonardo Pitalúa Salazar	Fecha:	05/09/2025
-----------------------	--------------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre> program FibonacciSerie implicit none integer :: n, i, a, b, temp ! Caleb pitalua write(*,*) 'Ingrese el número de términos (n): ' read(*,*) n a = 0 b = 1 if (n >= 1) then write(*,*) a end if if (n >= 2) then write(*,*) b end if do i = 3, n temp = a + b write(*,*) temp a = b b = temp end do end program FibonacciSerie </pre>	<p>10</p> <hr/> <p>Output:</p> <p>Ingrese el número de términos (n):</p> <pre> 0 1 1 2 3 5 8 13 21 34 </pre> <p>7</p> <hr/> <p>Output:</p> <p>Ingrese el número de términos (n):</p> <pre> 0 1 1 2 3 5 8 </pre>

--	--

Código en el lenguaje Pascal	Ejecución
<pre>{Caleb Leonardo Pitalúa Salazar} program Fibonacci; var n, i, a, b, temp: integer; begin write('Ingrese el número de términos (n): '); readln(n); a := 0; b := 1; if n >= 1 then write(a, ' '); if n >= 2 then write(b, ' '); for i := 3 to n do begin temp := a + b; write(temp, ' '); a := b; b := temp; end; writeln; end.</pre>	<pre>Free Pascal Compiler version 3.2.2+dfsg-32 [2024/01/05] for x86_64 Copyright (c) 1993-2021 by Florian Klaempfl and others Target OS: Linux for x86-64 Compiling main.pas Linking a.out 23 lines compiled, 0.0 sec Ingrese el número de términos (n): 10 0 1 1 2 3 5 8 13 21 34 ...Program finished with exit code 0 Press ENTER to exit console.</pre> <pre>Free Pascal Compiler version 3.2.2+dfsg-32 [2024/01/05] for x86_64 Copyright (c) 1993-2021 by Florian Klaempfl and others Target OS: Linux for x86-64 Compiling main.pas Linking a.out 23 lines compiled, 0.0 sec Ingrese el número de términos (n): 5 0 1 1 2 3 ...Program finished with exit code 0 Press ENTER to exit console.</pre>

--	--

Código en el lenguaje C/C++	Ejecución
<pre>#include <iostream> int main() { int n; std::cout << "Ingrese el número de términos (n): "; std::cin >> n; int a = 0; int b = 1; if (n >= 1) { std::cout << a << " "; } if (n >= 2) { std::cout << b << " "; } for (int i = 3; i <= n; ++i) { int temp = a + b; std::cout << temp << " "; a = b; b = temp; } std::cout << std::endl; return 0; }</pre>	<div>6</div> <p>Salida del programa</p> <pre>Ingrese el número de términos (n): 0 1 1 2 3 [Execution complete with exit code 0]</pre> <hr/> <div>4</div> <p>Salida del programa</p> <pre>Ingrese el número de términos (n): 0 1 1 2 [Execution complete with exit code 0]</pre>

--	--

Código en el lenguaje Java	Ejecución
<pre>import java.util.Scanner; // public class Fibonacci { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese el número de términos (n): "); int n = scanner.nextInt(); int a = 0; int b = 1; if (n >= 1) { System.out.print(a + " "); } if (n >= 2) { System.out.print(b + " "); } for (int i = 3; i <= n; i++) { int temp = a + b; System.out.print(temp + " "); a = b; b = temp; } System.out.println(); scanner.close(); } }</pre>	<pre>Ingrese el n?mero de t?rminos (n): 9 0 1 1 2 3 5 8 13 21 === Code Execution Successful ===</pre> <pre>Output</pre> <pre>Ingrese el n?mero de t?rminos (n): 12 0 1 1 2 3 5 8 13 21 34 55 89 === Code Execution Successful ===</pre>

--	--

Reporte de Examen práctico.

Problema #12: Leer una cadena y contar cuántas vocales y consonantes tiene. (15 puntos)


Nombre del alumno(a):	Caleb Leonardo Pitalúa Salazar	Fecha:	05/09/2025
-----------------------	--------------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre>! Caleb pitalua program ContarVocalesConsonantes implicit none character(len=100) :: cadena integer :: i, longitud, vocalCount, consonantCount character :: caracter write(*,*) 'Introduce una cadena:' read(*,*) cadena longitud = len_trim(cadena) vocalCount = 0 consonantCount = 0 do i = 1, longitud caracter = cadena(i:i) ! Convertir a mayúsculas para simplificar la comparación if (ichar(caracter) >= ichar('a') .and. ichar(caracter) <= ichar('z')) then caracter = char(ichar(caracter) - ichar('a') + ichar('A')) end if if (ichar(caracter) >= ichar('A') .and. ichar(caracter) <= ichar('Z')) then select case (caracter) case ('A', 'E', 'I', 'O', 'U') vocalCount = vocalCount + 1 case default consonantCount = consonantCount + 1 end select end if end do end program</pre>	<p><u>Programacion</u></p> <hr/> <p>Output:</p> <pre>Introduce una cadena: Número de vocales: 5 Número de consonantes: 7</pre> <p><u>Compilador</u></p> <hr/> <p>Output:</p> <pre>Introduce una cadena: Número de vocales: 4 Número de consonantes: 6</pre>

<pre> end if end do write(*,*) 'Número de vocales:', vocalCount write(*,*) 'Número de consonantes:', consonantCount end program ContarVocalesConsonantes </pre>	
---	--

Código en el lenguaje Pascal	Ejecución
<pre> {Caleb Leonardo Pitalúa Salazar} program ContarVocalesConsonantes; var cadena: string; i: integer; longitud: integer; vocalCount: integer; consonantCount: integer; caracter: char; begin writeln('Introduce una cadena:'); readln(cadena); longitud := length(cadena); vocalCount := 0; consonantCount := 0; for i := 1 to longitud do begin caracter := upcase(cadena[i]); // Convertir a mayúsculas para simplificar la comparación if (caracter >= 'A') and (caracter <= 'Z') then begin if (caracter = 'A') or (caracter = 'E') or (caracter = 'I') or (caracter = 'O') or (caracter = 'U') then begin </pre>	<pre> Free Pascal Compiler version 3.2.2+dfsg-32 [2024/01/ Copyright (c) 1993-2021 by Florian Klaempfl and othe Target OS: Linux for x86_64 Compiling main.pas Linking a.out 38 lines compiled, 0.0 sec Introduce una cadena: Esta es una cadena Número de vocales: 8 Número de consonantes: 7 Free Pascal Compiler version 3.2.2+dfsg-32 [202 Copyright (c) 1993-2021 by Florian Klaempfl and Target OS: Linux for x86_64 Compiling main.pas Linking a.out 38 lines compiled, 0.0 sec Introduce una cadena: Abecedario Número de vocales: 6 Número de consonantes: 4 </pre>

<pre> vocalCount := vocalCount + 1; end else begin consonantCount := consonantCount + 1; end; end; end; writeln('Número de vocales: ', vocalCount); writeln('Número de consonantes: ', consonantCount); readln; // Espera a que el usuario presione Enter para cerrar la consola end.</pre>	
--	--

Código en el lenguaje C/C++	Ejecución
<pre> #include <iostream> #include <string> #include <cctype> // Para la función toupper //Caleb Leonardo Pitalúa Salazar int main() { std::string cadena; int vocalCount = 0; int consonantCount = 0; std::cout << "Introduce una cadena: "; std::getline(std::cin, cadena); for (char caracter : cadena) { caracter = std::toupper(caracter); // Convertir a mayúsculas para simplificar la comparación if (caracter >= 'A' && caracter <= 'Z') {</pre>	<div>  </div> <p>Salida del programa</p> <pre> Introduce una cadena: Número de vocales: 8 Número de consonantes: 9 [Execution complete with exit code 0]</pre>

<pre> if (caracter == 'A' caracter == 'E' caracter == 'I' caracter == 'O' caracter == 'U') { vocalCount++; } else { consonantCount++; } } } std::cout << "Número de vocales: " << vocalCount << std::endl; std::cout << "Número de consonantes: " << consonantCount << std::endl; return 0; } </pre>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p style="color: #d9534f; font-size: small;">Constituyente</p> </div> <h4 style="color: #d9534f;">Salida del programa</h4> <div style="background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p>Introduce una cadena: Número de vocales: 5 Número de consonantes: 8</p> <p>[Execution complete with exit code 0]</p> </div>
--	--

Código en el lenguaje Java	Ejecución
<pre> import java.util.Scanner; //Caleb Leonardo Pitalua Salazar public class ContarVocalesConsonantes { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Introduce una cadena: "); String cadena = scanner.nextLine(); int vocalCount = 0; int consonantCount = 0; for (int i = 0; i < cadena.length(); i++) { char caracter = Character.toUpperCase(cadena.charAt(i)); // Convertir a mayúsculas </pre>	<div style="background-color: #2c3e50; color: white; padding: 10px;"> <h4 style="margin: 0;">Output</h4> <pre> Introduce una cadena: Taquimecanografia Número de vocales: 9 Número de consonantes: 8 === Code Execution Successful === Introduce una cadena: Emperatriz Número de vocales: 4 Número de consonantes: 6 === Code Execution Successful === </pre> </div>

```
        if (caracter >= 'A' && caracter <= 'Z') {  
            if (caracter == 'A' || caracter == 'E' ||  
caracter == 'I' || caracter == 'O' || caracter ==  
'U') {  
                vocalCount++;  
            } else {  
                consonantCount++;  
            }  
        }  
    }  
}  
  
    System.out.println("Número de vocales: " +  
vocalCount);  
    System.out.println("Número de  
consonantes: " + consonantCount);  
  
    scanner.close();  
}  
}
```

Reporte de Examen práctico.

Problema 18: Desarrollar un menu que permita actualizar los elementos de un arreglo desordenado para Crear, insertar, buscar y eliminar elementos. (30 puntos)

Nombre del alumno(a):	Caleb Leonardo Pitalúa Salazar	Fecha:	05/09/2025
-----------------------	--------------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre>! Caleb pitalua program ArregloDesordenado implicit none integer, parameter :: MAX_SIZE = 10 integer, dimension(MAX_SIZE) :: arreglo integer :: tamActual integer :: opcion, valor, posicion, i ! --- Subrutinas y Funciones --- ! Inicializa el arreglo call InicializarArreglo(arreglo, tamActual) ! Bucle principal del menú do while (opcion /= 0) print *, " print *, '--- MENU ARREGLO DESORDENADO ---' print *, '1. Crear/Reinicializar Arreglo' print *, '2. Insertar Elemento' print *, '3. Buscar Elemento' print *, '4. Eliminar Elemento' print *, '5. Mostrar Arreglo' print *, '0. Salir' print *, 'Ingrese su opcion: ' read (*,*) opcion</pre>	<pre>STDIN 1 2 25 2 144 3 144 0</pre>

```

select case (opcion)
  case (1)
    call InicializarArreglo(arreglo, tamActual)
  case (2)
    print *, 'Ingrese el valor a insertar: '
    read (*,*) valor
    call InsertarElemento(arreglo, tamActual, valor)
  case (3)
    print *, 'Ingrese el valor a buscar: '
    read (*,*) valor
    posicion = BuscarElemento(arreglo, tamActual,
valor)
    if (posicion /= -1) then
      print *, 'Elemento ', valor, ' encontrado en la
posicion ', posicion, '!'
    else
      print *, 'Elemento ', valor, ' no encontrado.'
    end if
  case (4)
    print *, 'Ingrese el valor a eliminar: '
    read (*,*) valor
    call EliminarElemento(arreglo, tamActual, valor)
  case (5)
    call MostrarArreglo(arreglo, tamActual)
  case (0)
    print *, 'Saliendo del programa.'
  case default
    print *, 'Opcion invalida. Intente de nuevo.'
end select
end do

```

contains

```

subroutine InicializarArreglo(arr, tam)
  integer, dimension(*) :: arr
  integer, intent(out) :: tam
  tam = 0
  print *, 'Arreglo inicializado.'
end subroutine InicializarArreglo

```

--- MENU ARREGLO DESORDENADO ---

1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir

Ingrese su opcion:
Arreglo inicializado.

--- MENU ARREGLO DESORDENADO ---

1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir

Ingrese su opcion:
Ingrese el valor a insertar:
Elemento 25 insertado.

--- MENU ARREGLO DESORDENADO ---

1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir

Ingrese su opcion:
Ingrese el valor a insertar:
Elemento 144 insertado.

```

subroutine MostrarArreglo(arr, tam)
  integer, dimension(*) :: arr
  integer, intent(in) :: tam
  integer :: k
  if (tam == 0) then
    print *, 'El arreglo esta vacio.'
    return
  end if
  write (*, '(A, *(I0, A))') 'Arreglo: [' , (arr(k), ' ', k=1,
tam-1), arr(tam), ']'
end subroutine MostrarArreglo

subroutine InsertarElemento(arr, tam, elemento)
  integer, dimension(*) :: arr
  integer, intent(inout) :: tam
  integer, intent(in) :: elemento
  if (tam < MAX_SIZE) then
    tam = tam + 1
    arr(tam) = elemento
    print *, 'Elemento ', elemento, ' insertado.'
  else
    print *, 'El arreglo esta lleno, no se puede insertar.'
  end if
end subroutine InsertarElemento

function BuscarElemento(arr, tam, elemento)
result(pos)
  integer, dimension(*) :: arr
  integer, intent(in) :: tam
  integer, intent(in) :: elemento
  integer :: pos, k
  pos = -1 ! No encontrado
  do k = 1, tam
    if (arr(k) == elemento) then
      pos = k
      return
    end if
  end do

```

```

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion:
Ingrese el valor a buscar:
Elemento      144 encontrado en la posicion      2 .

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion:
Saliendo del programa.

```

<pre> end function BuscarElemento subroutine EliminarElemento(arr, tam, elemento) integer, dimension(*) :: arr integer, intent(inout) :: tam integer, intent(in) :: elemento integer :: pos, k pos = BuscarElemento(arr, tam, elemento) if (pos /= -1) then ! Mover elementos hacia adelante para cubrir el espacio do k = pos, tam - 1 arr(k) = arr(k+1) end do tam = tam - 1 print *, 'Elemento ', elemento, ' eliminado.' else print *, 'Elemento ', elemento, ' no encontrado para eliminar.' end if end subroutine EliminarElemento </pre>	
--	--

Código en el lenguaje Pascal	Ejecución
------------------------------	-----------

```
{Caleb Leonardo Pitalúa Salazar}
program ArregloDesordenado;

const
    MAX_SIZE = 10;

type
    TArreglo = array[1..MAX_SIZE] of Integer;

var
    arreglo: TArreglo;
    tamActual: Integer;
    opcion: Integer;
    valor: Integer;
    posicion: Integer;

procedure InicializarArreglo(var arr: TArreglo; var
tam: Integer);
begin
    tam := 0; // El arreglo está vacío
    Writeln('Arreglo inicializado.');
```

```
end;

procedure MostrarArreglo(const arr: TArreglo;
tam: Integer);
var
    i: Integer;
begin
    if tam = 0 then
        begin
            Writeln('El arreglo esta vacio.');
```

```
        exit;
        end;
    Write('Arreglo: ');
    for i := 1 to tam do
        begin
```

```
--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 1
Arreglo inicializado.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 2
Ingrese el valor a insertar: 50
Elemento 50 insertado.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 2
Ingrese el valor a insertar: 20
Elemento 20 insertado.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 3
Ingrese el valor a buscar: 50
Elemento 50 encontrado en la posicion 1.
```

```
Write(arr[i]);
if i < tam then
  Write(' ');
end;
Writeln(']');
end;

procedure InsertarElemento(var arr: TArreglo; var
tam: Integer; elemento: Integer);
begin
  if tam < MAX_SIZE then
  begin
    inc(tam);
    arr[tam] := elemento;
    Writeln('Elemento ', elemento, ' insertado. ');
  end
  else
  begin
    Writeln('El arreglo esta lleno, no se puede
insertar. ');
  end;
end;

function BuscarElemento(const arr: TArreglo;
tam: Integer; elemento: Integer): Integer;
var
  i: Integer;
begin
  BuscarElemento := -1; // No encontrado
  for i := 1 to tam do
  begin
    if arr[i] = elemento then
    begin
      BuscarElemento := i; // Devuelve la posicion
(1-based)
    end;
  end;
```



```
end;
end;
end;

procedure EliminarElemento(var arr: TArreglo;
var tam: Integer; elemento: Integer);
var
  pos: Integer;
  i: Integer;
begin
  pos := BuscarElemento(arr, tam, elemento);
  if pos <> -1 then
    begin
      // Mover el ultimo elemento a la posicion del
      eliminado
      for i := pos to tam - 1 do
        begin
          arr[i] := arr[i + 1];
        end;
      dec(tam);
      Writeln('Elemento ', elemento, ' eliminado.');
```

end

```
    else
      begin
        Writeln('Elemento ', elemento, ' no encontrado
para eliminar.');
```

end;

```
  end;
end;

begin
  InicializarArreglo(arreglo, tamActual);

  repeat
    Writeln("");
    Writeln('--- MENU ARREGLO DESORDENADO ---
');
```

```
Writeln('1. Crear/Reinicializar Arreglo');
Writeln('2. Insertar Elemento');
Writeln('3. Buscar Elemento');
Writeln('4. Eliminar Elemento');
Writeln('5. Mostrar Arreglo');
Writeln('0. Salir');
Write('Ingrese su opcion: ');
Readln(opcion);

case opcion of
  1: InicializarArreglo(arreglo, tamActual);
  2:
    begin
      Write('Ingrese el valor a insertar: ');
      Readln(valor);
      InsertarElemento(arreglo, tamActual, valor);
    end;
  3:
    begin
      Write('Ingrese el valor a buscar: ');
      Readln(valor);
      posicion := BuscarElemento(arreglo,
tamActual, valor);
      if posicion <> -1 then
        Writeln('Elemento ', valor, ' encontrado en
la posicion ', posicion, '.')
      else
        Writeln('Elemento ', valor, ' no
encontrado.');
```

```
    end;
  4:
    begin
      Write('Ingrese el valor a eliminar: ');
      Readln(valor);
      EliminarElemento(arreglo, tamActual,
valor);
```

<pre> end; 5: MostrarArreglo(arreglo, tamActual); 0: Writeln('Saliendo del programa. '); else Writeln('Opcion invalida. Intente de nuevo. '); end; until opcion = 0; end. </pre>	
--	--

Código en el lenguaje C/C++	Ejecución
<pre> #include <iostream> #include <vector> //Caleb Leonardo Pitalúa Salazar // Para un arreglo C-style de tamaño fijo: // const int MAX_SIZE = 10; // int arreglo[MAX_SIZE]; // int tamActual = 0; void inicializarArreglo(std::vector<int>& arr) { arr.clear(); // Vacía el vector std::cout << "Arreglo inicializado." << std::endl; } void mostrarArreglo(const std::vector<int>& arr) { if (arr.empty()) { std::cout << "El arreglo esta vacio." << std::endl; </pre>	<pre> Arreglo inicializado. --- MENU ARREGLO DESORDENADO --- 1. Crear/Reinicializar Arreglo 2. Insertar Elemento 3. Buscar Elemento 4. Eliminar Elemento 5. Mostrar Arreglo 0. Salir Ingrese su opcion: Arreglo inicializado. --- MENU ARREGLO DESORDENADO --- 1. Crear/Reinicializar Arreglo 2. Insertar Elemento 3. Buscar Elemento 4. Eliminar Elemento 5. Mostrar Arreglo 0. Salir Ingrese su opcion: Ingrese el valor a insertar: Elemento 25 insertado. --- MENU ARREGLO DESORDENADO --- 1. Crear/Reinicializar Arreglo 2. Insertar Elemento 3. Buscar Elemento 4. Eliminar Elemento 5. Mostrar Arreglo 0. Salir Ingrese su opcion: Ingrese el valor a insertar: Elemento 15 insertado. --- MENU ARREGLO DESORDENADO --- 1. Crear/Reinicializar Arreglo 2. Insertar Elemento 3. Buscar Elemento 4. Eliminar Elemento 5. Mostrar Arreglo 0. Salir Ingrese su opcion: Ingrese el valor a buscar: Elemento 15 encontrado en la posicion 1. </pre>

```

    return;
}
std::cout << "Arreglo: [";
for (size_t i = 0; i < arr.size(); ++i) {
    std::cout << arr[i];
    if (i < arr.size() - 1) {
        std::cout << ", ";
    }
}
std::cout << "]" << std::endl;
}

void insertarElemento(std::vector<int>& arr, int
elemento) {
    // Si usaras un arreglo C-style con MAX_SIZE:
    // if (tamActual < MAX_SIZE) {
    //     arreglo[tamActual++] = elemento;
    //     std::cout << "Elemento " << elemento << "
insertado." << std::endl;
    // } else {
    //     std::cout << "El arreglo esta lleno, no se
puede insertar." << std::endl;
    // }

    arr.push_back(elemento); // std::vector maneja
el tamaño dinámicamente
    std::cout << "Elemento " << elemento << "
insertado." << std::endl;
}

int buscarElemento(const std::vector<int>& arr,
int elemento) {
    for (size_t i = 0; i < arr.size(); ++i) {
        if (arr[i] == elemento) {
            return i; // Devuelve la posicion (0-based)
        }
    }
}

```

```

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: Saliendo del programa.

[Execution complete with exit code 0]

```

```
}  
    return -1; // No encontrado  
}  
  
void eliminarElemento(std::vector<int>& arr, int  
elemento) {  
    int pos = buscarElemento(arr, elemento);  
    if (pos != -1) {  
        arr.erase(arr.begin() + pos); // Elimina el  
elemento en la posición  
        std::cout << "Elemento " << elemento << "  
eliminado." << std::endl;  
    } else {  
        std::cout << "Elemento " << elemento << " no  
encontrado para eliminar." << std::endl;  
    }  
}  
  
int main() {  
    std::vector<int> arreglo; // Usamos std::vector  
    int opcion;  
    int valor;  
    int posicion;  
  
    inicializarArreglo(arreglo);  
  
    do {  
        std::cout << "\n--- MENU ARREGLO  
DESORDENADO ---" << std::endl;  
        std::cout << "1. Crear/Reinicializar Arreglo"  
<< std::endl;  
        std::cout << "2. Insertar Elemento" <<  
std::endl;  
        std::cout << "3. Buscar Elemento" <<  
std::endl;
```

```

std::cout << "4. Eliminar Elemento" <<
std::endl;
std::cout << "5. Mostrar Arreglo" <<
std::endl;
std::cout << "0. Salir" << std::endl;
std::cout << "Ingrese su opcion: ";
std::cin >> opcion;

switch (opcion) {
    case 1:
        inicializarArreglo(arreglo);
        break;
    case 2:
        std::cout << "Ingrese el valor a insertar:
";

        std::cin >> valor;
        insertarElemento(arreglo, valor);
        break;
    case 3:
        std::cout << "Ingrese el valor a buscar: ";
        std::cin >> valor;
        posicion = buscarElemento(arreglo,
valor);
        if (posicion != -1) {
            std::cout << "Elemento " << valor << "
encontrado en la posicion " << posicion << "." <<
std::endl;
        } else {
            std::cout << "Elemento " << valor << "
no encontrado." << std::endl;
        }
        break;
    case 4:
        std::cout << "Ingrese el valor a eliminar:
";

        std::cin >> valor;

```

<pre> eliminarElemento(arreglo, valor); break; case 5: mostrarArreglo(arreglo); break; case 0: std::cout << "Saliendo del programa." << std::endl; break; default: std::cout << "Opcion invalida. Intente de nuevo." << std::endl; } } while (opcion != 0); return 0; } </pre>	
---	--

Código en el lenguaje Java	Ejecución
----------------------------	-----------

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class ArregloDesordenado {

    private List<Integer> arreglo;
    private Scanner scanner;

    public ArregloDesordenado() {
        arreglo = new ArrayList<>();
        scanner = new Scanner(System.in);
    }

    public void inicializarArreglo() {
        arreglo.clear(); // Vacía el ArrayList
        System.out.println("Arreglo inicializado.");
    }

    public void mostrarArreglo() {
        if (arreglo.isEmpty()) {
            System.out.println("El arreglo esta vacio.");
            return;
        }
        System.out.println("Arreglo: " +
arreglo.toString());
    }

    public void insertarElemento(int elemento) {
        arreglo.add(elemento); // Agrega al final
        System.out.println("Elemento " + elemento +
" insertado.");
    }

    public int buscarElemento(int elemento) {
```

```
Arreglo inicializado.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 2
Ingrese el valor a insertar: 12
Elemento 12 insertado.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 2
Ingrese el valor a insertar: 33
Elemento 33 insertado.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 3
Ingrese el valor a buscar: 33
Elemento 33 encontrado en la posicion 1.
```



```
// indexOf devuelve la primera ocurrencia del
// elemento, o -1 si no se encuentra
return arreglo.indexOf(elemento);
}

public void eliminarElemento(int elemento) {
    boolean eliminado =
    arreglo.remove(Integer.valueOf(elemento)); //
    remove(Object)
    if (eliminado) {
        System.out.println("Elemento " +
        elemento + " eliminado.");
    } else {
        System.out.println("Elemento " +
        elemento + " no encontrado para eliminar.");
    }
}

public void mostrarMenu() {
    int opcion;
    do {
        System.out.println("\n--- MENU ARREGLO
        DESORDENADO ---");
        System.out.println("1. Crear/Reinicializar
        Arreglo");
        System.out.println("2. Insertar Elemento");
        System.out.println("3. Buscar Elemento");
        System.out.println("4. Eliminar
        Elemento");
        System.out.println("5. Mostrar Arreglo");
        System.out.println("0. Salir");
        System.out.print("Ingrese su opcion: ");
        opcion = scanner.nextInt();

        switch (opcion) {
            case 1:
```

```
--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 3
Ingrese el valor a buscar: 12
Elemento 12 encontrado en la posicion 0.

--- MENU ARREGLO DESORDENADO ---
1. Crear/Reinicializar Arreglo
2. Insertar Elemento
3. Buscar Elemento
4. Eliminar Elemento
5. Mostrar Arreglo
0. Salir
Ingrese su opcion: 0
Saliendo del programa.

=== Code Execution Successful ===
```

```

        inicializarArreglo();
        break;
    case 2:
        System.out.print("Ingrese el valor a
insertar: ");
        int valorInsertar = scanner.nextInt();
        insertarElemento(valorInsertar);
        break;
    case 3:
        System.out.print("Ingrese el valor a
buscar: ");
        int valorBuscar = scanner.nextInt();
        int posicion =
buscarElemento(valorBuscar);
        if (posicion != -1) {
            System.out.println("Elemento " +
valorBuscar + " encontrado en la posicion " +
posicion + ".");
        } else {
            System.out.println("Elemento " +
valorBuscar + " no encontrado.");
        }
        break;
    case 4:
        System.out.print("Ingrese el valor a
eliminar: ");
        int valorEliminar = scanner.nextInt();
        eliminarElemento(valorEliminar);
        break;
    case 5:
        mostrarArreglo();
        break;
    case 0:
        System.out.println("Saliendo del
programa.");
        break;

```

```
default:
    System.out.println("Opcion invalida.
Intente de nuevo.");
    }
    } while (opcion != 0);
    scanner.close(); // Cerrar el scanner al salir
}

public static void main(String[] args) {
    ArregloDesordenado programa = new
ArregloDesordenado();
    programa.inicializarArreglo(); // Inicializa el
arreglo al iniciar el programa
    programa.mostrarMenu();
}
}
```