

Reporte de Examen práctico.

Problema 10: Multiplicación de matrices

Nombre del alumno(a):	Monica Ileana Ramírez Colin	Fecha:	05/09/2025
-----------------------	-----------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre> 11 programa MultiplicacionMatrices 12 ninguno implicito 13 14 entero :: filasA, columnasA, filasB, columnasB, i, j, k 15 entero, assignable :: matrizA (:,:), matrizB (:,:), matrizResultado (:,:) 16 17 print *, '=== MULTIPLICACIÓN DE MATRICES ===' 18 write (*, '(A)', avance = 'no') 'Filas de la matriz A: ' 19 leer *, filasA 20 escribir (*, '(A)', avance = 'no') 'Columnas de la matriz A: ' 21 leer *, columnasA 22 23 escribe (*, '(A)', avance = 'no') 'Filas de la matriz B: ' 24 leer *, filasB 25 escribir (*, '(A)', avance = 'no') 'Columnas de la matriz B: ' 26 leer *, columnasB 27 28 si (columnasA /= filasB) entonces 29 print *, 'Error: No se pueden multiplicar las matrices.' 30 print *, 'El número de columnas de A debe ser igual al número de filas de B.' 31 detener 32 terminar si 33 34 asignar (matrizA (filasA, columnasA)) 35 asignar (matrizB (filasB, columnasB)) 36 asignar (matrizResultado (filasA, columnasB)) 37 38 !Llenar matriz A 39 imprimir *, '' 40 print *, 'Ingrese los elementos de la matriz A:' 41 qhago i = 1, filasA 42 hacer j = 1, columnasA 43 escribir (*, '(A,II,A,II,A)', avanzar = 'no') 'A[' , i - 1, '][', j - 1, ']: ' 44 leer *, matrizA (i, j) 45 fin hacer 46 fin qhago 47 48 !Llenar matriz B 49 imprimir *, '' 50 print *, 'Ingrese los elementos de la matriz B:' 51 qhago i = 1, filasB 52 hacer j = 1, columnasB 53 escribir (*, '(B,II,B,II,B)', avanzar = 'no') 'B[' , i - 1, '][', j - 1, ']: ' 54 leer *, matrizB (i, j) 55 fin hacer 56 fin qhago 57 58 !Calcular el resultado 59 do i = 1, filasA 60 do j = 1, columnasB 61 k = 0 62 do k = 1, filasB 63 k = k + matrizA(i,k) * matrizB(k,j) 64 end do 65 matrizResultado(i,j) = k 66 end do 67 end do 68 69 !Desasignar matrices 70 desasignar (matrizA) 71 desasignar (matrizB) 72 desasignar (matrizResultado) 73 74 fin del programa MultiplicacionMatrices </pre>	<pre> === MULTIPLICACION DE MATRICES === Filas de la matriz A: 2 Columnas de la matriz A: 2 Filas de la matriz B: 2 Columnas de la matriz B: 2 Ingrese los elementos de la matriz A: A[0][0]: 2 A[0][1]: 5 A[1][0]: 4 A[1][1]: 7 Ingrese los elementos de la matriz B: B[0][0]: 8 B[0][1]: 5 B[1][0]: 1 B[1][1]: 24 === RESULTADOS === Matriz A: 2 5 4 7 Matriz B: 8 5 1 24 Matriz Resultado (A x B): 21 130 39 188 </pre>

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

```

1 program MultiplicacionMatrices;
2 uses
3   SysUtils;
4
5 var
6   filasA, columnasA, filasB, columnasB, i, j, k: Integer;
7   matrizA, matrizB, matrizResultado: array of array of Integer;
8
9 begin
10  Writeln('=== MULTIPLICACIÓN DE MATRICES ===');
11  Write('Filas de la matriz A: ');
12  Readln(filasA);
13  Write('Columnas de la matriz A: ');
14  Readln(columnasA);
15
16  Write('Filas de la matriz B: ');
17  Readln(filasB);
18  Write('Columnas de la matriz B: ');
19  Readln(columnasB);
20
21  if columnasA <> filasB then
22  begin
23    Writeln('Error: No se pueden multiplicar las matrices. ');
24    Writeln('El número de columnas de A debe ser igual al número de filas de B. ');
25    Exit;
26  end;
27
28  SetLength(matrizA, filasA, columnasA);
29  SetLength(matrizB, filasB, columnasB);
30  SetLength(matrizResultado, filasA, columnasB);
31
32  Writeln('');
33  Writeln('Ingrese los elementos de la matriz A:');
34  for i := 0 to filasA - 1 do
35  begin
36    for j := 0 to columnasA - 1 do
37    begin
38      Write('A[', i, '][', j, ']: ');
39      Readln(matrizA[i][j]);
40    end;
41  end;
42
43  Writeln('');
44  Writeln('Ingrese los elementos de la matriz B:');
45  for i := 0 to filasB - 1 do
46  begin
47    for j := 0 to columnasB - 1 do
48    begin
49      Write('B[', i, '][', j, ']: ');
50      Readln(matrizB[i][j]);
51    end;
52  end;
53
54  for i := 0 to filasA - 1 do
55  begin
56    for j := 0 to columnasB - 1 do
57    begin
58      matrizResultado[i][j] := 0;
59      for k := 0 to columnasA - 1 do
60      begin
61        matrizResultado[i][j] := matrizResultado[i][j] + (matrizA[i][k] * matrizB[k][j]);
62      end;
63    end;
64  end;
65
66  Writeln('');
67  Writeln('=== RESULTADOS ===');
68
69  Writeln('Matriz A:');
70  for i := 0 to filasA - 1 do
71  begin
72    for j := 0 to columnasA - 1 do
73    begin
74      Write(matrizA[i][j]:4, ' ');
75    end;
76    Writeln('');
77  end;
78
79  Writeln('');
80
81  Writeln('Matriz B:');
82  for i := 0 to filasB - 1 do
83  begin
84    for j := 0 to columnasB - 1 do
85    begin
86      Write(matrizB[i][j]:4, ' ');
87    end;
88    Writeln('');
89  end;
90
91  Writeln('');
92  Writeln('Matriz Resultado (A x B):');
93  for i := 0 to filasA - 1 do
94  begin
95    for j := 0 to columnasB - 1 do
96    begin
97      Write(matrizResultado[i][j]:6, ' ');
98    end;
99    Writeln('');
100  end;
101
102  Readln;
103 end.

```

```

102 lines compiled, 0.1 sec
=== MULTIPLICACIÓN DE MATRICES ===
Filas de la matriz A: 2
Columnas de la matriz A: 2
Filas de la matriz B: 2
Columnas de la matriz B: 2

Ingrese los elementos de la matriz A:
A[0][0]: 5
A[0][1]: 7
A[1][0]: 8
A[1][1]: 4

Ingrese los elementos de la matriz B:
B[0][0]: 1
B[0][1]: 5
B[1][0]: 6
B[1][1]: 9

=== RESULTADOS ===
Matriz A:
  5  7
  8  4

Matriz B:
  1  5
  6  9

Matriz Resultado (A x B):
 47 86
 32 74

```

Código en el lenguaje C/C++

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int filasA, columnasA, filasB, columnasB;
6
7     printf("=== MULTIPLICACIÓN DE MATRICES ===\n");
8     printf("Filas de la matriz A: ");
9     scanf("%d", &filasA);
10    printf("Columnas de la matriz A: ");
11    scanf("%d", &columnasA);
12
13    printf("Filas de la matriz B: ");
14    scanf("%d", &filasB);
15    printf("Columnas de la matriz B: ");
16    scanf("%d", &columnasB);
17
18    if (columnasA != filasB) {
19        printf("Error: No se pueden multiplicar las matrices.\n");
20        printf("El número de columnas de A debe ser igual al número de filas de B.\n");
21        return 1;
22    }
23
24    int **matrizA = (int **)malloc(filasA * sizeof(int));
25    int **matrizB = (int **)malloc(filasB * sizeof(int));
26    int **matrizResultado = (int **)malloc(filasA * sizeof(int));
27
28    for (int i = 0; i < filasA; i++) {
29        for (int j = 0; j < columnasA; j++) {
30            matrizA[i][j] = (int *)malloc(columnasA * sizeof(int));
31        }
32    }
33
34    for (int i = 0; i < filasB; i++) {
35        for (int j = 0; j < columnasB; j++) {
36            matrizB[i][j] = (int *)malloc(columnasB * sizeof(int));
37        }
38    }
39
40    printf("\nIngrese los elementos de la matriz A:\n");
41    for (int i = 0; i < filasA; i++) {
42        for (int j = 0; j < columnasA; j++) {
43            printf("A[%d][%d]: ", i, j);
44            scanf("%d", &matrizA[i][j]);
45        }
46    }
47
48    printf("\nIngrese los elementos de la matriz B:\n");
49    for (int i = 0; i < filasB; i++) {
50        for (int j = 0; j < columnasB; j++) {
51            printf("B[%d][%d]: ", i, j);
52            scanf("%d", &matrizB[i][j]);
53        }
54    }
55
56    for (int i = 0; i < filasA; i++) {
57        for (int j = 0; j < columnasB; j++) {
58            for (int k = 0; k < columnasA; k++) {
59                matrizResultado[i][j] += matrizA[i][k] * matrizB[k][j];
60            }
61        }
62    }
63
64    printf("\n=== RESULTADOS ===\n");
65
66    printf("Matriz A:\n");
67    for (int i = 0; i < filasA; i++) {
68        for (int j = 0; j < columnasA; j++) {
69            printf("%4d ", matrizA[i][j]);
70        }
71        printf("\n");
72    }
73
74    printf("Matriz B:\n");
75    for (int i = 0; i < filasB; i++) {
76        for (int j = 0; j < columnasB; j++) {
77            printf("%4d ", matrizB[i][j]);
78        }
79        printf("\n");
80    }
81
82    printf("Matriz Resultado (A x B):\n");
83    for (int i = 0; i < filasA; i++) {
84        for (int j = 0; j < columnasB; j++) {
85            printf("%4d ", matrizResultado[i][j]);
86        }
87        printf("\n");
88    }
89
90    for (int i = 0; i < filasA; i++) {
91        free(matrizA[i]);
92    }
93    free(matrizResultado);
94
95    for (int i = 0; i < filasB; i++) {
96        free(matrizB[i]);
97    }
98    free(matrizB);
99}
```

Ejecución

```
=== MULTIPLICACIÓN DE MATRICES ===
Filas de la matriz A: 2
Columnas de la matriz A: 2
Filas de la matriz B: 2
Columnas de la matriz B: 2

Ingrese los elementos de la matriz A:
A[0][0]: 1
A[0][1]: 4
A[1][0]: 5
A[1][1]: 2

Ingrese los elementos de la matriz B:
B[0][0]: 8
B[0][1]: 4
B[1][0]: 5
B[1][1]: 5

=== RESULTADOS ===
Matriz A:
1  4
5  2

Matriz B:
8  4
5  5

Matriz Resultado (A x B):
28  24
50  30
```

```

75     for (int j = 0; j < columnasB; j++) {
76         printf("%4d ", matrizB[i][j]);
77     }
78     printf("\n");
79     printf("\n");
80
81     printf("Matriz Resultado (A x B):\n");
82     for (int i = 0; i < filasA; i++) {
83         for (int j = 0; j < columnasB; j++) {
84             printf("%6d ", matrizResultado[i][j]);
85         }
86         printf("\n");
87     }
88
89     for (int i = 0; i < filasA; i++) {
90         free(matrizA[i]);
91         free(matrizResultado[i]);
92     }
93     for (int i = 0; i < filasB; i++) {
94         free(matrizB[i]);
95     }
96     free(matrizA);
97     free(matrizB);
98     free(matrizResultado);
99
100    return 0;
101
102 }

```

Código en el lenguaje Java

```

// MultiplicacionMatrices.java
import java.util.Scanner;

public class MultiplicacionMatrices {

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

        System.out.println("===== INICIALIZACION DE MATRICES =====");
        System.out.println("Filas de la matriz A: ");
        int filasA = scanner.nextInt();
        System.out.println("Columnas de la matriz A: ");
        int columnasA = scanner.nextInt();

        System.out.println("Filas de la matriz B: ");
        int filasB = scanner.nextInt();
        System.out.println("Columnas de la matriz B: ");
        int columnasB = scanner.nextInt();

        if (columnasA != filasB) {
            System.out.println("Error: No se pueden multiplicar las matrices.");
            System.out.println("Filas de la matriz A: " + filasA + " y columnas de la matriz A: " + columnasA);
            System.out.println("Filas de la matriz B: " + filasB + " y columnas de la matriz B: " + columnasB);
            return;
        }

        int[][] matrizA = new int[filasA][columnasA];
        int[][] matrizB = new int[filasB][columnasB];
        int[][] matrizResultado = new int[filasA][columnasB];

        System.out.println("Ingrese los elementos de la matriz A:");
        for (int i = 0; i < filasA; i++) {
            for (int j = 0; j < columnasA; j++) {
                matrizA[i][j] = scanner.nextInt();
            }
        }

        System.out.println("Ingrese los elementos de la matriz B:");
        for (int i = 0; i < filasB; i++) {
            for (int j = 0; j < columnasB; j++) {
                matrizB[i][j] = scanner.nextInt();
            }
        }

        multiplicarMatrices(matrizA, matrizB, matrizResultado);

        System.out.println("===== RESULTADO =====");
        System.out.println("Matriz A:");
        mostrarMatriz(matrizA);
        System.out.println("Matriz B:");
        mostrarMatriz(matrizB);
        System.out.println("Matriz Resultado (A x B):");
        mostrarMatriz(matrizResultado);
        scanner.close();
    }

    public static void multiplicarMatrices(int[][] A, int[][] B, int[][] resultado) {
        for (int i = 0; i < A.length; i++) {
            for (int j = 0; j < B[0].length; j++) {
                resultado[i][j] = 0;
                for (int k = 0; k < A[0].length; k++) {
                    resultado[i][j] += A[i][k] * B[k][j];
                }
            }
        }
    }

    public static void mostrarMatriz(int[][] matriz) {
        for (int i = 0; i < matriz.length; i++) {
            for (int j = 0; j < matriz[0].length; j++) {
                System.out.print(matriz[i][j] + " ");
            }
            System.out.println();
        }
    }
}

```

Ejecución

```

===== INICIALIZACION DE MATRICES =====
Filas de la matriz A: 2
Columnas de la matriz A: 3
Filas de la matriz B: 3
Columnas de la matriz B: 3

Ingrese los elementos de la matriz A:
A[0][0]: 2
A[0][1]: 5
A[0][2]: 8
A[1][0]: 5
A[1][1]: 5
A[1][2]: 5

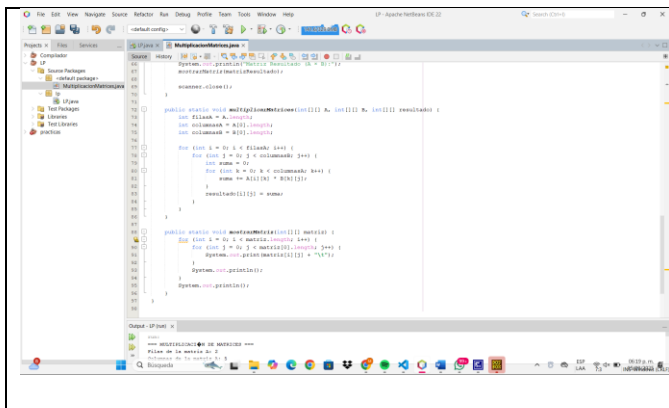
Ingrese los elementos de la matriz B:
B[0][0]: 5
B[0][1]: 5
B[0][2]: 5
B[1][0]: 5
B[1][1]: 5
B[1][2]: 5

===== RESULTADO =====
Matriz A:
2 5 8
5 5 5

Matriz B:
5 5 5
5 5 5
5 5 5

Matriz Resultado (A x B):
40 115
40 136

```



Problema 11: Verificar si un número o palabra se lee igual al derecho y al revés.

Nombre del alumno(a):	Monica Ileana Ramírez Colín	Fecha:	05/09/2025
-----------------------	-----------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
<pre> 37 Programa VerificadorPalindromo 38 ninguno implicito 39 carácter (len = 100) :: entrada 40 lógico :: resultado 41 42 print *, '=== VERIFICADOR DE PALÍNDROMOS ===' 43 print *, 'Ingrese un número o palabra: ' 44 leer (*, '(A)') entrada 45 46 resultado = esPalindromo (trim (entrada)) 47 48 si (resultado) entonces 49 print *, trim (entrada), 'ES un palíndromo' 50 demás 51 print *, trim (entrada), 'NO es un palíndromo' 52 terminar si 53 54 contiene 55 56 función lógica esPalindromo (texto) 57 carácter (len = *) , intención (in) :: texto 58 carácter (len = len (texto)) :: textolimpio 59 entero :: i , j , longitud 60 61 textolimpio = texto 62 hacer i = 1 , len (textolimpio) 63 si (textolimpio (i : i) .eq. 'A' .v. textolimpio (i : i) .eq. '2') entonces </pre>	

```

12 carácter ( len = * ), intención ( in ) :: texto
36 carácter ( len = len ( texto ) ) :: textolimpio
42 entero :: i , j , longitud
43
44
39 textolimpio = texto
40 hacer i = 1 , len ( textolimpio )
41 si ( textolimpio ( i : i ) >= 'A' .y. textolimpio ( i : i ) <= 'Z' ) entonces
34 textolimpio ( i : i ) = char ( ichar ( textolimpio ( i : i ) ) + 32 )
35 de lo contrario si ( textolimpio ( i : i ) == ' ' ) entonces
48 textolimpio ( i : i ) = ''
38 terminar si
37 fin hacer
18
46 textolimpio = ajustar ( textolimpio )
45 longitud = len_trim ( textolimpio )
19
4 esPalindromo = .true.
3 chago i = 1 , longitud / 2 ?
2 j = longitud - i + 1
7 si ( textolimpio ( i : i ) /= textolimpio ( j : j ) ) entonces
6 esPalindromo = .falso .
5 salida
1 terminar si
10 fin hacer
9 función final esPalindromo
8
47 Fin del programa VerificadorPalindromo

```

```

=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra:
2002
2002 ES un palíndromo

...Program finished with exit code 0
Press ENTER to exit console.

```

```

=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra:
reconocer
reconocer ES un palíndromo

...Program finished with exit code 0
Press ENTER to exit console.

```

Código en el lenguaje Pascal

```

14 programa VerificadorPalindromo ;
13 usos
12 Utilidades del sistema , Crt ;
27
26 función esPalindromo ( texto : string ) : boolean ;
25 variedad
30 textolimpio : cadena ;
29 i , longitud : entero ;
28 comenzar
10 textolimpio := LowerCase ( StringReplace ( texto , ' ' , '' , [ rfReplaceAll ] ) ) ;
31
32 longitud := Longitud ( textolimpio ) ;
33 para i := 1 a longitud div 2 hacer
34 comenzar
35 si textolimpio [ i ] <> textolimpio [ longitud - i + 1 ] entonces
36 comenzar
37 esPalindromo := falso ;
38 salida ;
39 fin ;
1 fin ;
2 esPalindromo := verdadero ;
3 fin ;
4 variedad
5 entrada : cadena ;
6 comenzar
7 ClrScr ;
8 WriteLn ( ' === VERIFICADOR DE PALÍNDROMOS === ' ) ;

```

Ejecución

```

=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra: 22
22 ' ES un palíndromo

```

```

=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra: RECONOCER
RECONOCER' ES un palíndromo

```

```

31 longitud := longitud ( textolimpio );
32 para i := 1 a longitud div 2 hacer
33     comenzar
34         si textolimpio [ i ] <> textolimpio [ longitud - i + 1 ] entonces
35             comenzar
36                 esPalindromo := falso ;
37             Salida ;
38             fin ;
39         fin ;
40     fin ;
41 esPalindromo := verdadero ;
42 fin ;
43
44 variedad
45 entrada : cadena ;
46 comenzar
47     ClrScr ;
48     Writeln ( ' === VERIFICADOR DE PALÍNDROMOS === ' );
49     Write ( ' Ingrese un número o palabra: ' );
50     Readln ( entrada );
51
52     si esPalindromo ( entrada ) entonces
53         Writeln ( entrada , ' ' ' ES un palindromo ' );
54     demás
55         Writeln ( entrada , ' ' ' NO es un palindromo ' );
56
57     Leerln ;
58 fin ;

```

```

=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra: GATO
GATO' NO es un palíndromo

```

```

=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra: 13
13' NO es un palíndromo

```

Código en el lenguaje C/C++

```

1 #include <stdio.h>
2 #include <ctype.h>
3 #include <string.h>
4 #include <stdbool.h>
5
6 bool esPalindromo(char texto[]) {
7     char textolimpio[100];
8     int i, j = 0;
9     int longitud;
10
11     for (i = 0; texto[i] != '\0'; i++) {
12         if (texto[i] != ' ') {
13             textolimpio[j++] = tolower(texto[i]);
14         }
15     }
16     textolimpio[j] = '\0';
17
18     longitud = strlen(textolimpio);
19     for (i = 0; i < longitud / 2; i++) {
20         if (textolimpio[i] != textolimpio[longitud - i - 1]) {
21             return false;
22         }
23     }
24     return true;
25 }
26
27 int main() {
28     char entrada[100];
29
30     printf("=== VERIFICADOR DE PALÍNDROMOS ===\n");
31     printf("Ingrese un número o palabra: ");
32     fgets(entrada, sizeof(entrada), stdin);
33
34     entrada[strcspn(entrada, "\n")] = '\0';
35
36     if (esPalindromo(entrada)) {
37         printf("ES un palindromo\n", entrada);
38     } else {
39         printf("NO es un palindromo\n", entrada);
40     }
41
42     return 0;
43 }

```

Ejecución

```

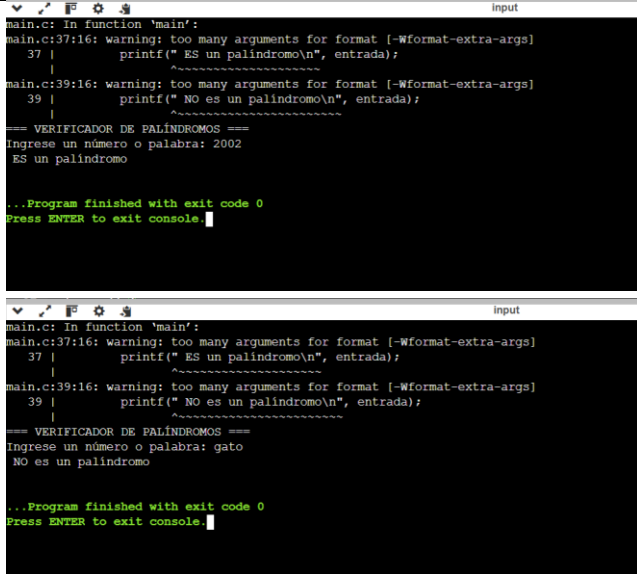
main.c: In function 'main':
main.c:37:16: warning: too many arguments for format [-Wformat-extra-args]
   37 |         printf(" ES un palindromo\n", entrada);
      |         ~~~~~^
main.c:39:16: warning: too many arguments for format [-Wformat-extra-args]
   39 |         printf(" NO es un palindromo\n", entrada);
      |         ~~~~~^
=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra: 12
NO es un palindromo

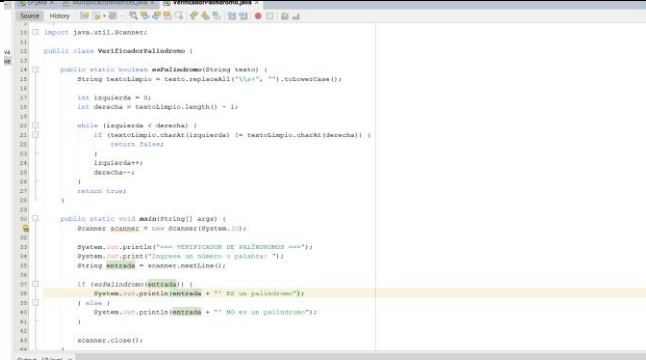
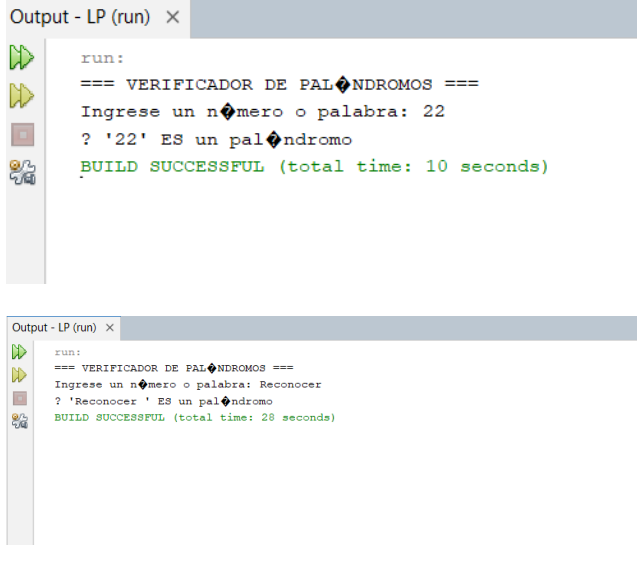
...Program finished with exit code 0
Press ENTER to exit console.

main.c: In function 'main':
main.c:37:16: warning: too many arguments for format [-Wformat-extra-args]
   37 |         printf(" ES un palindromo\n", entrada);
      |         ~~~~~^
main.c:39:16: warning: too many arguments for format [-Wformat-extra-args]
   39 |         printf(" NO es un palindromo\n", entrada);
      |         ~~~~~^
=== VERIFICADOR DE PALÍNDROMOS ===
Ingrese un número o palabra: reconocer
ES un palindromo

...Program finished with exit code 0
Press ENTER to exit console.

```

	
--	--

Código en el lenguaje Java	Ejecución
	

	<p>Output - LP (run) ×</p> <pre>run: === VERIFICADOR DE PALINDROMOS === Ingrese un número o palabra: 13 ? '13' NO es un palindromo BUILD SUCCESSFUL (total time: 5 seconds)</pre>
	<p>Output - LP (run) ×</p> <pre>run: === VERIFICADOR DE PALINDROMOS === Ingrese un número o palabra: gato ? 'gato' NO es un palindromo BUILD SUCCESSFUL (total time: 11 seconds)</pre>

Problema 9: Calcular la suma de dos matrices

Nombre del alumno(a):	Monica Ileana Ramírez Colín	Fecha:	05/09/2025
-----------------------	-----------------------------	--------	------------

Código en el lenguaje Fortran	Ejecución
-------------------------------	-----------

```

65 programa SumaMatrices
66     ninguno_implicito
67
68     entero :: filas , columnas , i , j
69     entero , assignable :: matrizA ( : , : ) , matrizB ( : , : ) , matrizSuma ( : , : )
36
2     escribe ( * , * ) 'Ingrese el número de filas: '
3     leer ( * , * ) filas
4     escribe ( * , * ) 'Ingrese el número de columnas: '
5     leer ( * , * ) columnas
7
8     asignar ( matrizA ( filas , columnas ) )
9     asignar ( matrizB ( filas , columnas ) )
10    asignar ( matrizSuma ( filas , columnas ) )
11
12    escribe ( * , * ) 'Ingrese los elementos de la primera matriz:'
13    {hago i = 1 , filas?
14        { j = 1 , columnas?
52            escribir ( * , * , '(A, I2, A, I2, A)' 'Elemento [', i - 1 , '][', j - 1 , ']: '
53            leer ( * , * ) matrizA ( i , j )
54        fin hacer
55    }
56 fin hacer
57
58    escribe ( * , * ) 'Ingrese los elementos de la segunda matriz:'
59    {hago i = 1 , filas?
60        { j = 1 , columnas?
61            escribir ( * , * , '(A, I2, A, I2, A)' 'Elemento [', i - 1 , '][', j - 1 , ']: '
62            leer ( * , * ) matrizB ( i , j )

```

```

62            leer ( * , * ) matrizB ( i , j )
63        fin hacer
64    }
65 fin hacer
66
67    {hago i = 1 , filas?
68        { j = 1 , columnas?
69            matrizSuma ( i , j ) = matrizA ( i , j ) + matrizB ( i , j )
70        fin hacer
71    }
72 fin hacer
73
74    escribe ( * , * ) '=== RESULTADOS ==='
75    escribe ( * , * ) 'Matriz A:'
36    llamar a mostrarMatriz ( matrizA , filas , columnas )
35
76    escribe ( * , * ) 'Matriz B:'
77    llamar a mostrarMatriz ( matrizB , filas , columnas )
78
68    escribe ( * , * ) 'Suma (A + B):'
69    llamar a mostrarMatriz ( matrizSuma , filas , columnas )
70
12    desasignar ( matrizA )
52    desasignar ( matrizB )
53    desasignar ( matrizSuma )
54
55 contiene
56
57 subrutina mostrarMatriz ( matriz , nfilas , ncolumnas )
58     entero , intención ( en ) :: nfilas , ncolumnas
59     entero , intención ( en ) :: matriz ( nfilas , ncolumnas )
60     entero :: i , j
61
62     {hago i = 1 , nfilas?
63         { j = 1 , ncolumnas?
64             escribir ( * , * , '(I6, A)' , avanzar = 'no' ) matriz ( i , j ) , ' '
65         fin hacer
66     }
67     escribir ( * , * )
68     fin de subrutina mostrarMatriz
69
70 fin del programa SumaMatrices

```

```

8     escribe ( * , * ) 'Matriz B:'
7     llamar a mostrarMatriz ( matrizB , filas , columnas )
6
68    escribe ( * , * ) 'Suma (A + B):'
11    llamar a mostrarMatriz ( matrizSuma , filas , columnas )
10
9    desasignar ( matrizA )
12    desasignar ( matrizB )
52    desasignar ( matrizSuma )
53
54 contiene
55
56 subrutina mostrarMatriz ( matriz , nfilas , ncolumnas )
57     entero , intención ( en ) :: nfilas , ncolumnas
58     entero , intención ( en ) :: matriz ( nfilas , ncolumnas )
59     entero :: i , j
60
61     {hago i = 1 , nfilas?
62         { j = 1 , ncolumnas?
63             escribir ( * , * , '(I6, A)' , avanzar = 'no' ) matriz ( i , j ) , ' '
64         fin hacer
65     }
66     escribir ( * , * )
67     fin de subrutina mostrarMatriz
68
69 fin del programa SumaMatrices

```

```

aporte
Ingrese el número de filas:
2
Ingrese el número de columnas:
1
Ingrese los elementos de la primera matriz:
Elemento [ 0][ 0]:
5
Elemento [ 1][ 0]:
4
Ingrese los elementos de la segunda matriz:
Elemento [ 0][ 0]:
8
Elemento [ 1][ 0]:
7
=== RESULTADOS ===
Matriz A:
5
4
Matriz B:
8
7
Suma (A + B):
13
11

```

Código en el lenguaje Pascal

Ejecución

```

1 program SumaMatrices;
2
3 uses
4   SysUtils;
5
6 type
7   TMatriz = array of array of integer;
8
9 procedure ingresarMatriz(var matriz: TMatriz; filas, columnas: integer);
10 var
11   i, j: integer;
12 begin
13   for i := 0 to filas - 1 do
14     begin
15       for j := 0 to columnas - 1 do
16         begin
17           Write('Elemento [', i, '][', j, ']: ');
18           ReadLn(matriz[i][j]);
19         end;
20       end;
21     end;
22 end;
23
24 procedure sumarMatrices(matrizA, matrizB: TMatriz; var matrizSuma: TMatriz; filas, columnas: integer);
25 var
26   i, j: integer;
27 begin
28   for i := 0 to filas - 1 do
29     begin
30       for j := 0 to columnas - 1 do
31         begin
32           matrizSuma[i][j] := matrizA[i][j] + matrizB[i][j];
33         end;
34       end;
35     end;
36 end;

```

```

31   matrizSuma[i][j] := matrizA[i][j] + matrizB[i][j];
32 end;
33 end;
34 end;
35 end;
36
37 procedure mostrarMatriz(matriz: TMatriz; filas, columnas: integer);
38 var
39   i, j: integer;
40 begin
41   for i := 0 to filas - 1 do
42     begin
43       for j := 0 to columnas - 1 do
44         begin
45           Write(matriz[i][j]:4, ' ');
46         end;
47       end;
48       WriteLn;
49     end;
50 end;
51
52 var
53   filas, columnas, i: integer;
54   matrizA, matrizB, matrizSuma: TMatriz;
55
56 begin
57   Write('Ingrese el número de filas: ');
58   ReadLn(filas);
59   Write('Ingrese el número de columnas: ');
60   ReadLn(columnas);
61   SetLength(matrizA, filas, columnas);
62   SetLength(matrizB, filas, columnas);
63   SetLength(matrizSuma, filas, columnas);
64 end;

```

```

55 begin
56   Write('Ingrese el número de filas: ');
57   ReadLn(filas);
58   Write('Ingrese el número de columnas: ');
59   ReadLn(columnas);
60
61   SetLength(matrizA, filas, columnas);
62   SetLength(matrizB, filas, columnas);
63   SetLength(matrizSuma, filas, columnas);
64
65   WriteLn('Ingrese los elementos de la primera matriz:');
66   ingresarMatriz(matrizA, filas, columnas);
67
68   WriteLn('Ingrese los elementos de la segunda matriz:');
69   ingresarMatriz(matrizB, filas, columnas);
70
71   sumarMatrices(matrizA, matrizB, matrizSuma, filas, columnas);
72
73   WriteLn('=== RESULTADOS ===');
74   WriteLn('Matriz A:');
75   mostrarMatriz(matrizA, filas, columnas);
76
77   WriteLn('Matriz B:');
78   mostrarMatriz(matrizB, filas, columnas);
79
80   WriteLn('Suma (A + B):');
81   mostrarMatriz(matrizSuma, filas, columnas);
82
83   SetLength(matrizA, 0, 0);
84   SetLength(matrizB, 0, 0);
85   SetLength(matrizSuma, 0, 0);
86 end.

```

```

Target: GNU Linux for amd64
Compiling main.pas
main.pas(52,20) Note: Local variable "i" not used
Linking *.out
15 lines compiled, 0.1 sec
1 note(s) issued
Ingrese el número de filas: 2
Ingrese el número de columnas: 2
Ingrese los elementos de la primera matriz:
Elemento [0][0]: 1
Elemento [0][1]: 4
Elemento [1][0]: 7
Elemento [1][1]: 1
Ingrese los elementos de la segunda matriz:
Elemento [0][0]: 5
Elemento [0][1]: 8
Elemento [1][0]: 2
Elemento [1][1]: 4
=== RESULTADOS ===
Matriz A:
1 4
7 1
Matriz B:
5 8
2 6
Suma (A + B):
6 12
9 7

```

Código en el lenguaje C/C++

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void ingresarMatriz(int **matriz, int filas, int columnas) {
5     for (int i = 0; i < filas; i++) {
6         for (int j = 0; j < columnas; j++) {
7             printf("Elemento [%d][%d]: ", i, j);
8             scanf("%d", &matriz[i][j]);
9         }
10    }
11 }
12
13 void sumarMatrices(int **matrizA, int **matrizB, int **matrizSuma, int filas, int columnas) {
14     for (int i = 0; i < filas; i++) {
15         for (int j = 0; j < columnas; j++) {
16             matrizSuma[i][j] = matrizA[i][j] + matrizB[i][j];
17         }
18     }
19 }
20
21 void mostrarMatriz(int **matriz, int filas, int columnas) {
22     for (int i = 0; i < filas; i++) {
23         for (int j = 0; j < columnas; j++) {
24             printf("%d\t", matriz[i][j]);
25         }
26         printf("\n");
27     }
28     printf("\n");
29 }
30
31 int main() {
32     int filas, columnas;
33
34     printf("Ingrese el número de filas: ");
35     scanf("%d", &filas);
36     printf("Ingrese el número de columnas: ");
37     scanf("%d", &columnas);
38
39     int **matrizA = (int **)malloc(filas * sizeof(int *));
40     int **matrizB = (int **)malloc(filas * sizeof(int *));
41     int **matrizSuma = (int **)malloc(filas * sizeof(int *));
42
43     for (int i = 0; i < filas; i++) {
44         matrizA[i] = (int *)malloc(columnas * sizeof(int));
45         matrizB[i] = (int *)malloc(columnas * sizeof(int));
46         matrizSuma[i] = (int *)malloc(columnas * sizeof(int));
47     }
48
49     printf("\nIngrese los elementos de la primera matriz:\n");
50     ingresarMatriz(matrizA, filas, columnas);
51
52     printf("\nIngrese los elementos de la segunda matriz:\n");
53     ingresarMatriz(matrizB, filas, columnas);
54
55     sumarMatrices(matrizA, matrizB, matrizSuma, filas, columnas);
56
57     printf("\n=== RESULTADOS ===\n");
58     printf("Matriz A:\n");
59     mostrarMatriz(matrizA, filas, columnas);
60
61     printf("Matriz B:\n");
62     mostrarMatriz(matrizB, filas, columnas);
63
64     printf("Suma (A + B):\n");
65     mostrarMatriz(matrizSuma, filas, columnas);
66
67     for (int i = 0; i < filas; i++) {
68         free(matrizA[i]);
69         free(matrizB[i]);
70         free(matrizSuma[i]);
71     }
72
73     free(matrizA);
74     free(matrizB);
75     free(matrizSuma);
76
77     return 0;
78 }

```

Ejecución

```

Input
Ingrese el número de filas: 2
Ingrese el número de columnas: 2

Ingrese los elementos de la primera matriz:
Elemento [0][0]: 2
Elemento [0][1]: 4
Elemento [1][0]: 7
Elemento [1][1]: 1

Ingrese los elementos de la segunda matriz:
Elemento [0][0]: 6
Elemento [0][1]: 9
Elemento [1][0]: 8
Elemento [1][1]: 5

=== RESULTADOS ===
Matriz A:
2   4
7   1

Matriz B:
6   9
8   5

Suma (A + B):
8   13
15   6

...Program finished with exit code 0
Press ENTER to exit console

```

Código en el lenguaje Java

```

1  public class Matrices {
2      public static void main(String[] args) {
3          Scanner scanner = new Scanner(System.in);
4
5          System.out.print("Ingresa el número de filas: ");
6          int filas = scanner.nextInt();
7
8          System.out.print("Ingresa el número de columnas: ");
9          int columnas = scanner.nextInt();
10
11          int[][] matrizA = new int[filas][columnas];
12          int[][] matrizB = new int[filas][columnas];
13
14          System.out.print("Ingresa los elementos de la primera matriz:\n");
15          ingresarMatrizA(matrizA, scanner);
16
17          System.out.print("Ingresa los elementos de la segunda matriz:\n");
18          ingresarMatrizB(matrizB, scanner);
19
20          mostrarMatriz(matrizA, matrizB, scanner);
21
22          System.out.print("Matriz A:\n");
23          mostrarMatriz(matrizA);
24
25          System.out.print("Matriz B:\n");
26          mostrarMatriz(matrizB);
27
28          System.out.print("Suma (A + B):\n");
29          mostrarMatriz(matrizSuma);
30          scanner.close();
31      }
32
33      public static void ingresarMatrizA(int[][] matriz, Scanner scanner) {
34          for (int i = 0; i < matriz.length; i++) {
35              for (int j = 0; j < matriz[i].length; j++) {
36                  System.out.print("Ingresa el elemento " + i + " x " + j + ": ");
37                  matriz[i][j] = scanner.nextInt();
38              }
39          }
40      }
41
42      public static void ingresarMatrizB(int[][] matriz, Scanner scanner) {
43          for (int i = 0; i < matriz.length; i++) {
44              for (int j = 0; j < matriz[i].length; j++) {
45                  System.out.print("Ingresa el elemento " + i + " x " + j + ": ");
46                  matriz[i][j] = scanner.nextInt();
47              }
48          }
49      }
50
51      public static void mostrarMatrizA(int[][] matriz) {
52          for (int i = 0; i < matriz.length; i++) {
53              for (int j = 0; j < matriz[i].length; j++) {
54                  System.out.print(matriz[i][j] + " ");
55              }
56              System.out.println();
57          }
58      }
59
60      public static void mostrarMatrizB(int[][] matriz) {
61          for (int i = 0; i < matriz.length; i++) {
62              for (int j = 0; j < matriz[i].length; j++) {
63                  System.out.print(matriz[i][j] + " ");
64              }
65              System.out.println();
66          }
67      }
68
69      public static void mostrarMatrizSuma(int[][] matrizA, int[][] matrizB) {
70          int[][] matrizSuma = new int[matrizA.length][matrizA[0].length];
71          for (int i = 0; i < matrizA.length; i++) {
72              for (int j = 0; j < matrizA[0].length; j++) {
73                  matrizSuma[i][j] = matrizA[i][j] + matrizB[i][j];
74              }
75          }
76          mostrarMatriz(matrizSuma);
77      }
78  }

```

Ejecución

```

1  Ingresa el número de filas: 2
2  Ingresa el número de columnas: 2
3
4  Ingresa los elementos de la primera matriz:
5  Elemento [0][0]: 1
6  Elemento [0][1]: 3
7  Elemento [1][0]: 4
8  Elemento [1][1]: 6
9
10 Ingresa los elementos de la segunda matriz:
11 Elemento [0][0]: 4
12 Elemento [0][1]: 5
13 Elemento [1][0]: 8
14 Elemento [1][1]: 9
15
16 Suma (A + B):
17
18 Matriz A:
19 1 3
20 4 6
21
22 Matriz B:
23 4 5
24 8 9
25
26 Suma (A + B):
27 5 8
28 12 15
29
30 RESULTADO: (total líneas: 17 segundos)

```