**POST LABORATORIO PRACTICA Nro : 4**

|  |  |
| --- | --- |
| **Estudiante 1** | **César Barreto** |
| **Estudiante 2** |  |

**NO ESCRIBA EN ESTE ESPACIO**

**Leer con detenimiento cada una de las siguientes instrucciones para el envío de la práctica:**

|  |
| --- |
| **Instrucciones para el envío de la práctica** :  Uno de los integrantes del grupo debe subir a Módulo 7, en la tarea que se ha habilitado para la práctica correspondiente a la semana, lo siguiente :   1. Este archivo .doc, que contenga por cada problema planteado :  * El análisis (identificar entrada, proceso, salidas) * El algoritmo en pseudocódigo. * El código fuente ( código en lenguaje Pascal- copia/pega del .pas) * Las capturas de pantallas del resultado de la compilación * Las capturas de pantallas de las corridas (ejecución) del programa, donde se visualicen los datos de entrada y los resultados producidos por el programa.  1. La HOJA DE CONTROL DE PRACTICAS con la autoevaluación de la práctica de la semana. (esta hoja la debe enviar cada estudiante por separado, pero uno solo debe enviar la prácica)   Importante :  NO CREE UNA CARPETA COMPRIMIDA PARA EL ENVIO DE LA PRÁCTICA |

**EJERCICIO 1**

1. Dada una matriz N X M (N y M no exceder de 10), elaborar un programa en Pascal que haga el recorrido en forma de espiral. El recorrido debe hacerse en sentido horario o en sentido anti horario, e ir llenando las celdas en una secuencia de números que inician en 1 hasta NxM. En el ejemplo A se muestran el llenado de una matriz 4x5 en sentido horario. (10 ptos)

**ANALISIS:**

|  |  |  |
| --- | --- | --- |
| **Entradas** | **Procesos** | **Salida** |
| Tamaño de la matriz:  N, M.  Sentido de llenado de la matriz:  Sentido. | Pedir los datos de entrada.  Realizar el llenado de la matriz, dependiendo si es en sentido horario o anti horario.  Mostrar los datos de salida | Mostrar la matriz llena. |

**ALGORITMO EN PSEUDOCÓDIGO:**

**Cabecera:** Llenado en espiral;

**Declaraciones**

Matriz: Arreglo [1..10, 1..10] entero;

Sentido: Carácter;

i,j, Numero, LimI\_N, LimiF\_N, LimI\_M, LimF\_N: Entero;

**Cuerpo**

**Inicio**

i <- 0;

j <- 0;

Numero <- 1

Repetir

Escribir (“Ingrese el tamaño deseado de la matriz. Las filas o las columnas no puede ser mayor de 10: ”);

Leer (N, M);

Si (N>10) o (M>10) Entonces

Escribir (“La cantidad de filas o la cantidad de columnas excede el límite. Vuelva a intentarlo.”);

Fin-Si;

Hasta (N <= 10) Y (M <= 10);

Fin-RH;

Para i, 1, N, +1 Hacer

Para j, 1, M, +1 Hacer

Matriz [N,M]<- 0;

Fin-RP;

Fin-RP;

Escribir (“Sentido de llenado de su matriz:”);

Escribir (“1. Sentido horario”);

Escribir (“2. Sentido anti horario”);

Leer (Sentido);

En caso de Sentido Hacer

“1”:

LimI\_N<- 1;

LimF\_N <- N

LimI\_M <- 1;

LimF\_M <- M;

Repetir Mientras (Numero <> N\*M) Hacer

Repetir para i, LimI\_M, LimF\_M, +1 Hacer

Si (Numero <= N\*M) Entonces

Matriz [LimI\_N, i] <- Numero;

Numero<- Numero + 1;

Fin-Si;

Fin-RP

LimI\_N <- LimI\_N + 1;

Repetir para i, LimI\_N, LimF\_N, +1 Hacer

Si (Numero <= N\*M) Entonces

Matriz [i, LimF\_M] <- Numero;

Numero<- Numero + 1;

Fin-Si;

Fin-RP;

LimF\_M <- LimF\_M – 1;

Repetir para i, LimF\_M, LimI\_M, -1 Hacer

Si (Numero <= N\*M) Entonces

Matriz [LimF\_N, i] <- Numero;

Numero<- Numero + 1;

Fin-SI;

Fin-RP;

LimF\_N<- LimF\_N-1;

Repetir para i, LimF\_N, LimI\_N, -1, Hacer

Si (Numero <= N\*M) Entonces

Matriz [i, LimI\_M] <- Numero;

Numero<- Numero + 1;

Fin-Si;

Fin-RP;

LimI\_M<- LimI\_M +1;

Fin-RM;

Para i, 1, N, +1 Hacer

Para j, 1, M, +1 Hacer

Write (Matriz [i,j], ‘ ‘);

Fin-RP;

Writeln ();

Fin-RP;

Fin-“1”;

“2”:

LimI\_N <- 1;

LimF\_N <- N;

LimI\_M <- M;

LimF\_M <- 1;

Repetir Mientras (Numero <= N\*M) Hacer

Repetir Para i, LimI\_M, LimF\_M, -1, Hacer

Si (Numero <= N\*M) Entonces

Matriz [LimI\_N, i] <- Numero;

Numero <- Numero + 1;

Fin-Si;

Fin-RP;

LimI\_N <- LimI\_N + 1;

Repetir Para i, LimI\_N, LimF\_N, +1, Hacer

Si (Numero <= N\*M) Entonces

Matriz [i, LimF\_M] <- Numero;

Numero <- Numero + 1;

Fin-Si;

Fin-RP;

LimF\_M <- LimF\_M + 1;

Repetir Para i, LimF\_M, LimI\_M, +1, Hacer

Si (Numero <= N\*M) Entonces

Matriz [LimF\_N, i] <- Numero;

Numero <- Numero + 1;

Fin-Si;

Fin-RP

LimF\_N := LimF\_N-1;

Repetir Para i, LimF\_N, LimI\_N, -1, Hacer

Si (Numero <= N\*M) Entonces

Matriz [i, LimI\_M] <- Numero;

Numero <- Numero + 1;

Fin-Si;

Fin-RP;

LimI\_M := LimI\_M -1;

Fin-RM;

Repetir Para i, 1, N, +1 Hacer

Repetir Para j, 1, M, +1 Hacer

Escribir (Matriz [i,j], “ “);

Fin-RP;

Escribir ();

Fin-RP;

Fin- “2”;

Fin-Caso;

**Fin.**

**PROGRAMA FUENTE:**

Program LlenadodeMatriz;

Uses

Crt;

Var

Sentido: Char;

i,j Numero, LimI\_N, LimF\_N, LimI\_M, LimF\_M, N, M: integer;

Matriz: array [1..10 , 1..10] Of integer;

Begin

i := 0;

j:= 0;

Numero := 1;

Repeat

clrscr;

Writeln ('Ingrese el tamanio deseado de la matriz. Las filas o las columnas no pueden ser mayor de 10: ‘);

Readln (N, M);

If (N>10) Or (M>10) Then

Writeln ('La cantidad de filas o la cantidad de columnas excede el limite. Vuelva a intentarlo.');

Until (N <= 10) And (M <= 10);

For i:= 1 To N Do

For j:= 1 To M Do

Matriz [N,M] := 0;

Writeln ('Sentido de llenado de su matriz:');

Writeln ('1. Sentido horario');

Writeln ('2. Sentido anti horario');

Readln (Sentido);

Case Sentido Of

'1':

Begin

LimI\_N := 1;

LimF\_N := N;

LimI\_M := 1;

LimF\_M := M;

While (Numero <= N\*M) Do

Begin

For i:= LimI\_M To LimF\_M Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [LimI\_N, i] := Numero;

Numero := Numero + 1;

End;

End;

LimI\_N := LimI\_N + 1;

For i:= LimI\_N To LimF\_N Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [i, LimF\_M] := Numero;

Numero := Numero + 1;

End;

End;

LimF\_M := LimF\_M - 1;

For i:= LimF\_M Downto LimI\_M Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [LimF\_N, i] := Numero;

Numero := Numero + 1;

End;

End;

LimF\_N := LimF\_N-1;

For i:= LimF\_N Downto LimI\_N Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [i, LimI\_M] := Numero;

Numero := Numero + 1;

End;

End;

LimI\_M := LimI\_M +1;

End;

clrscr;

For i:= 1 To N Do

Begin

For j:= 1 To M Do

Write (Matriz [i,j], ' ');

Writeln ();

End;

End;

'2':

Begin

LimI\_N := 1;

LimF\_N := N;

LimI\_M := M;

LimF\_M := 1;

While (Numero <= N\*M) Do

Begin

For i:= LimI\_M Downto LimF\_M Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [LimI\_N, i] := Numero;

Numero := Numero + 1;

End;

End;

LimI\_N := LimI\_N + 1;

For i:= LimI\_N To LimF\_N Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [i, LimF\_M] := Numero;

Numero := Numero + 1;

End;

End;

LimF\_M := LimF\_M + 1;

For i:= LimF\_M To LimI\_M Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [LimF\_N, i] := Numero;

Numero := Numero + 1;

End;

End;

LimF\_N := LimF\_N-1;

For i:= LimF\_N Downto LimI\_N Do

Begin

If (Numero <= N\*M)Then

Begin

Matriz [i, LimI\_M] := Numero;

Numero := Numero + 1;

End;

End;

LimI\_M := LimI\_M -1;

End;

clrscr;

For i:= 1 To N Do

Begin

For j:= 1 To M Do

Write (Matriz [i,j], ' ');

Writeln ();

End;

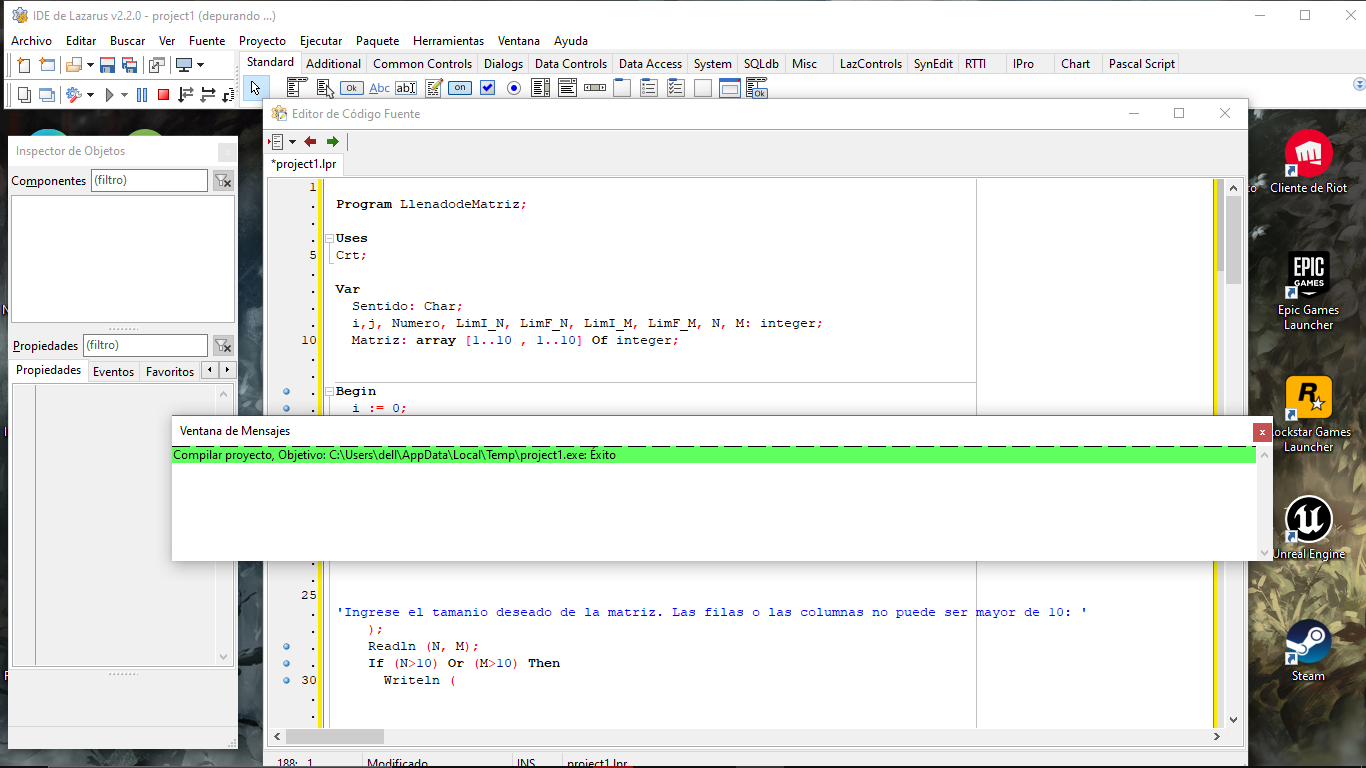
End;

End;

Readkey;

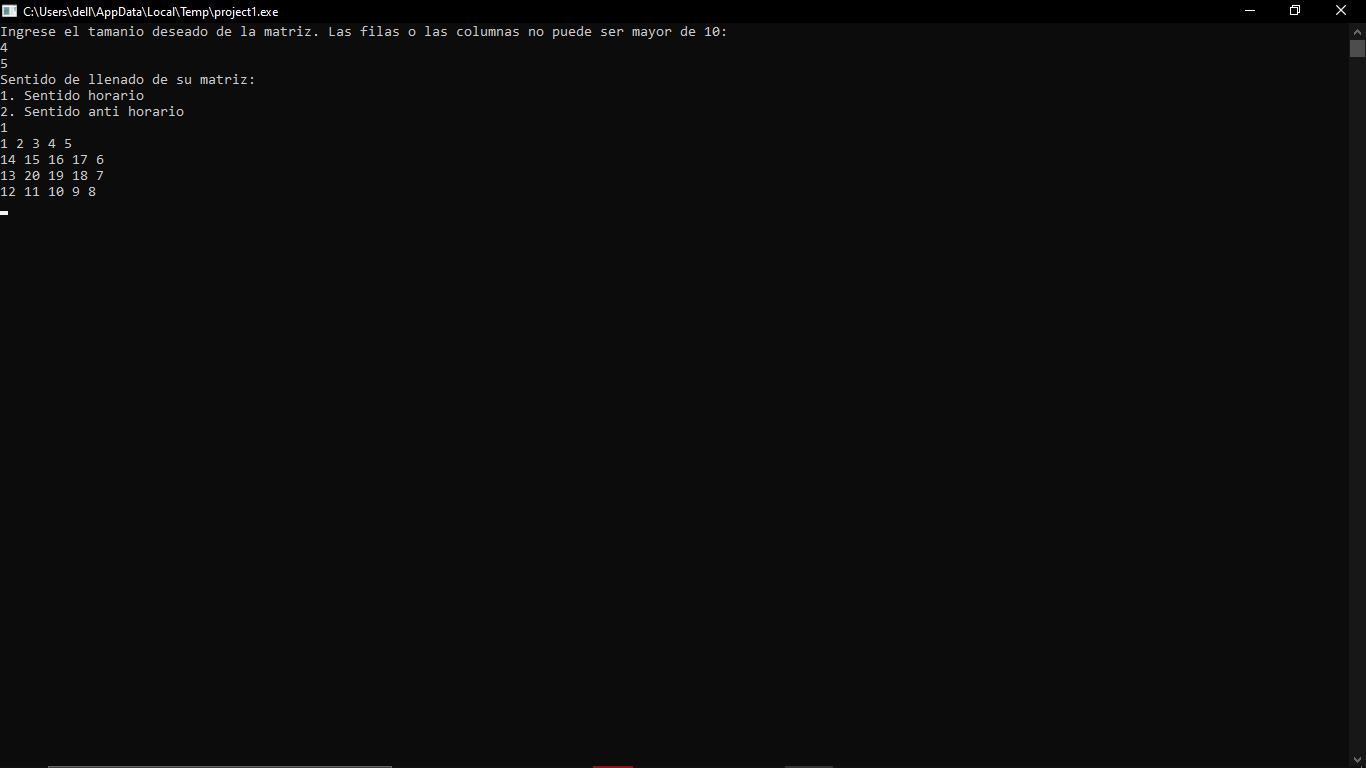
End.

**CAPTURA DE PANTALLA CON RESULTADO DE COMPILACIÓN:**

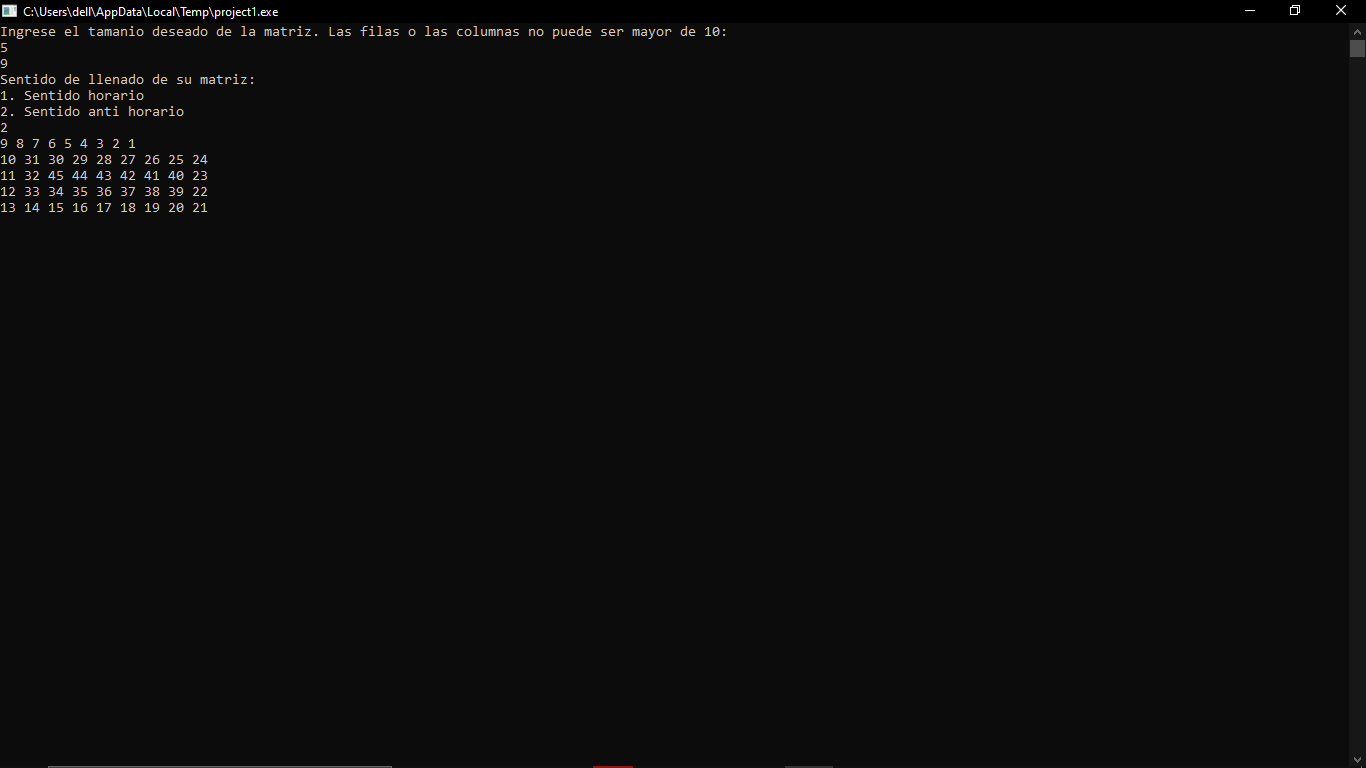


**CAPTURA DE PANTALLAS CON RESULTADO DE EJECUCIÓN (CORRIDAS)**

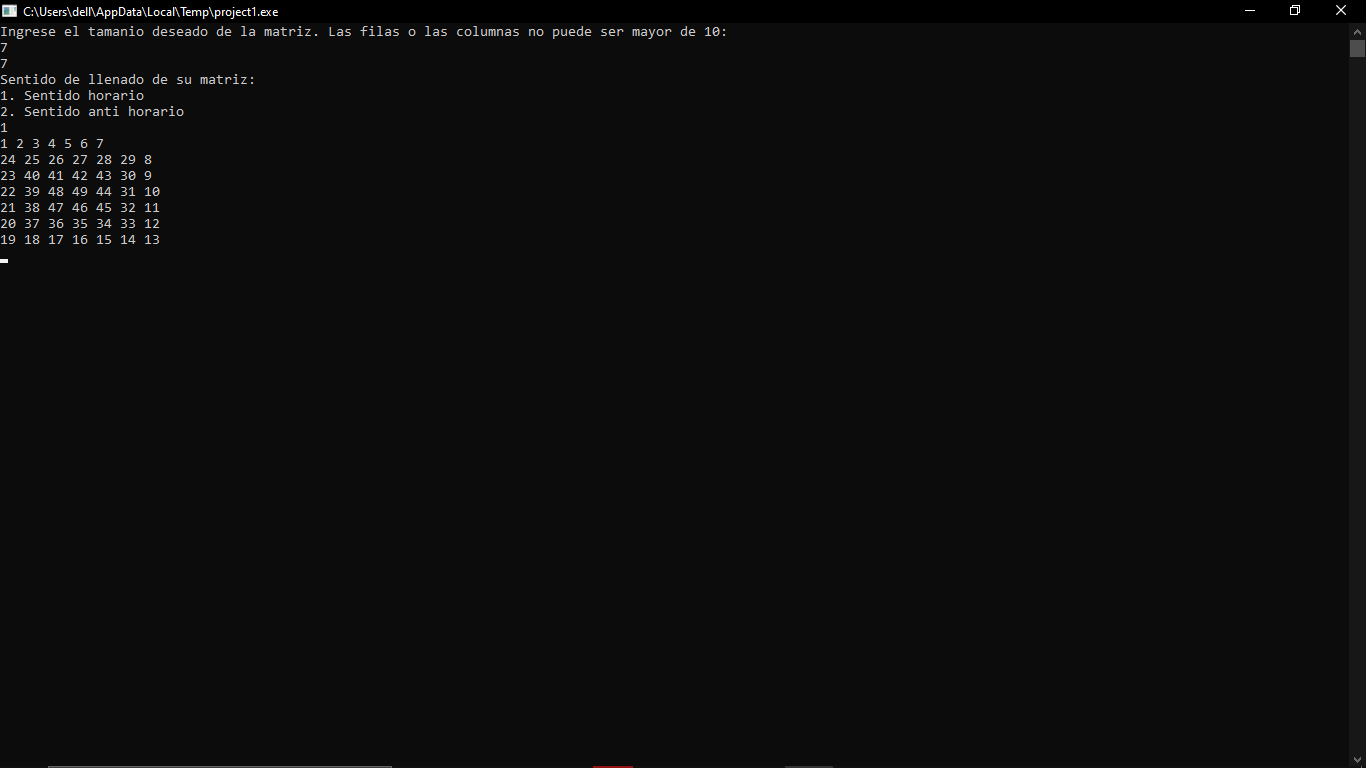
**Ejemplo 1:**



**Ejemplo 2:**



**Ejemplo 3:**



**EJERCICIO 2**

**ANALISIS:**

|  |  |  |
| --- | --- | --- |
| **Entradas** | **Procesos** | **Salida** |
|  |  |  |

**ALGORITMO EN PSEUDOCÓDIGO:**

xxxxxx

**PROGRAMA FUENTE:**

xxxxx

**CAPTURA DE PANTALLA CON RESULTADO DE COMPILACIÓN:**

xxxxxx

**CAPTURA DE PANTALLAS CON RESULTADO DE EJECUCIÓN (CORRIDAS)**

xxxxx

**EJERCICIO 3**

**ANALISIS:**

|  |  |  |
| --- | --- | --- |
| **Entradas** | **Procesos** | **Salida** |
|  |  |  |

**ALGORITMO EN PSEUDOCÓDIGO:**

xxxxxx

**PROGRAMA FUENTE:**

xxxxx

**CAPTURA DE PANTALLA CON RESULTADO DE COMPILACIÓN:**

xxxxxx

**CAPTURA DE PANTALLAS CON RESULTADO DE EJECUCIÓN (CORRIDAS)**

xxxxx