**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.  Limpiar toda la matriz  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

//Inicializacion de las variables

i := 0;

j:= 0;

Numero := 1;

//Validando los datos de entrada

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);

//Limpieza de la matriz

For i:= 1 To N Do

For j:= 1 To M Do

Matriz [N,M] := 0;

//Seleccion del sentido de llenado de la matriz

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

Writeln ('1. Sentido horario');

Writeln ('2. Sentido anti horario');

Readln (Sentido);

Case Sentido Of

'1':

Begin

//Indicando cuales son los primeros limites o bordes de la matriz

LimI\_N := 1;

LimF\_N := N;

LimI\_M := 1;

LimF\_M := M;

//Se inicia el proceso de llenado de la matriz, en sentido horario

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;

//Mostrando la matriz rellena

clrscr;

For i:= 1 To N Do

Begin

For j:= 1 To M Do

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

Writeln ();

End;

End;

'2':

Begin

//Indicando cuales son los primeros limites o bordes de la matriz

LimI\_N := 1;

LimF\_N := N;

LimI\_M := M;

LimF\_M := 1;

//Se inicia el proceso de llenado de la matriz, en sentido anti horario

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;

//Mostrando la matriz rellena

clrscr;

For i:= 1 To N Do

Begin

For j:= 1 To M Do

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

Writeln ();

End;

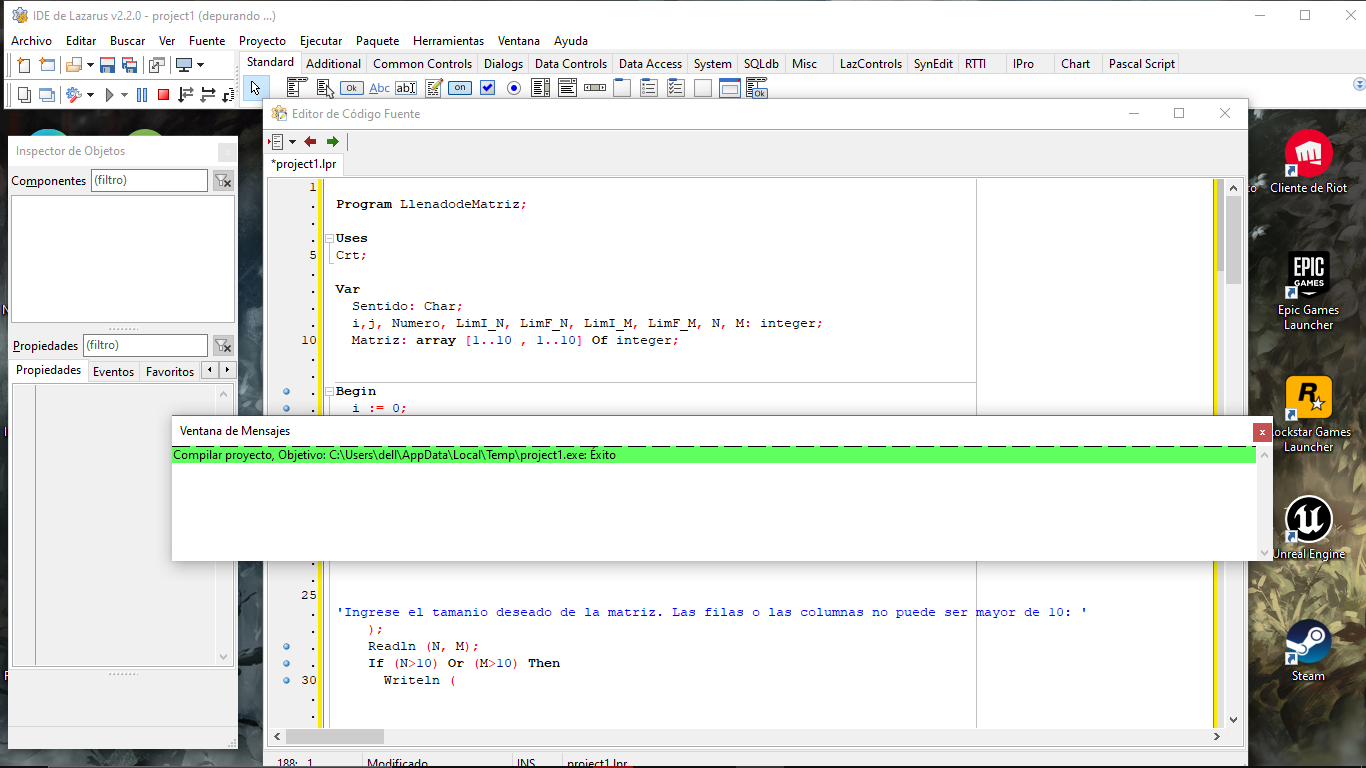
End;

End;

Readkey;

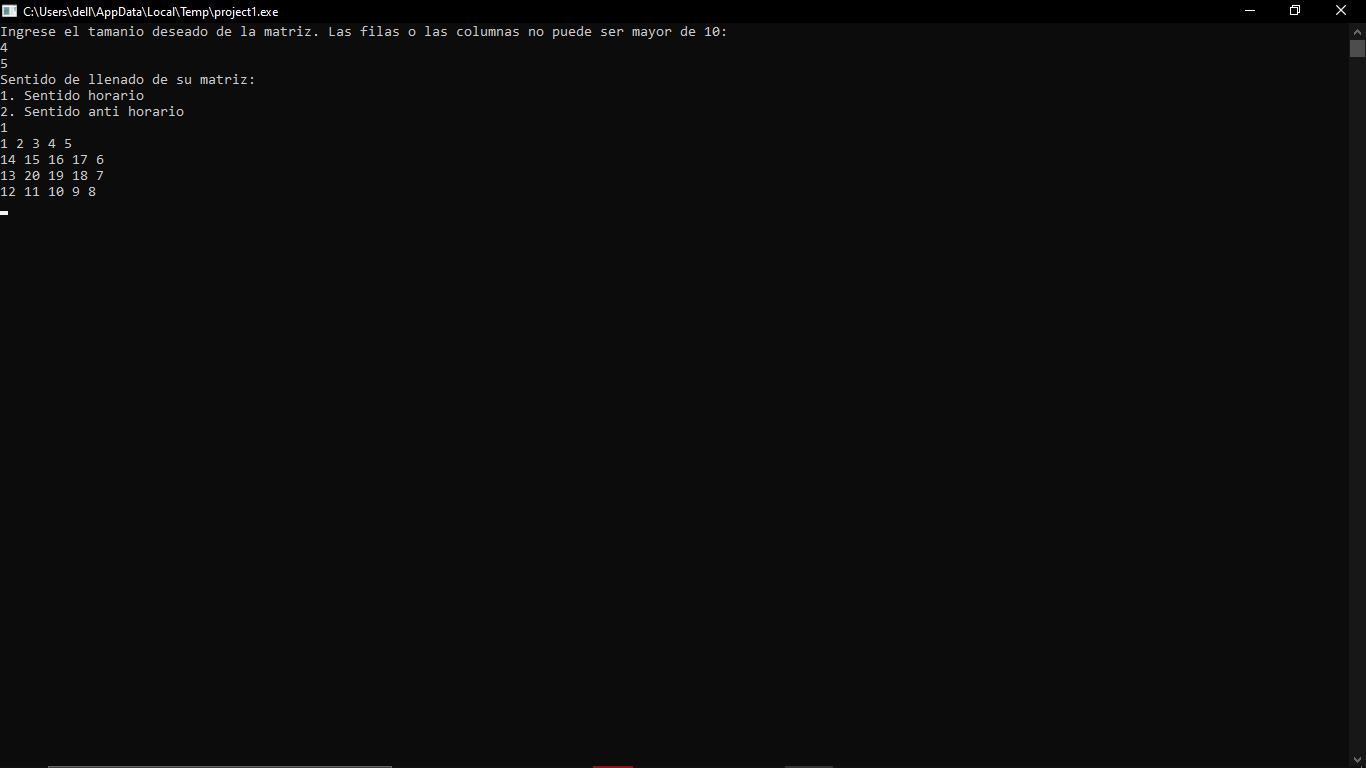
End.

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

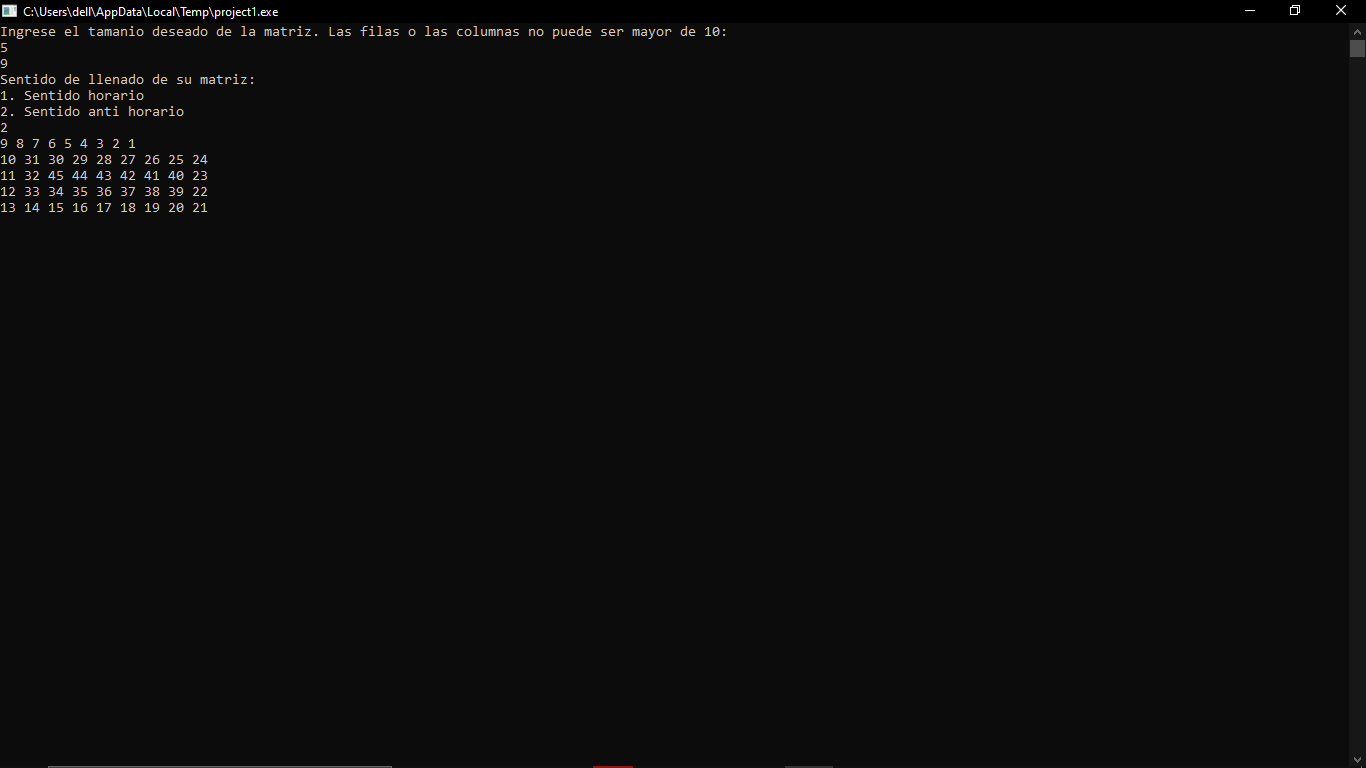


**CAPTURA DE PANTALLAS CON RESULTADO DE EJECUCIÓN (CORRIDAS)**

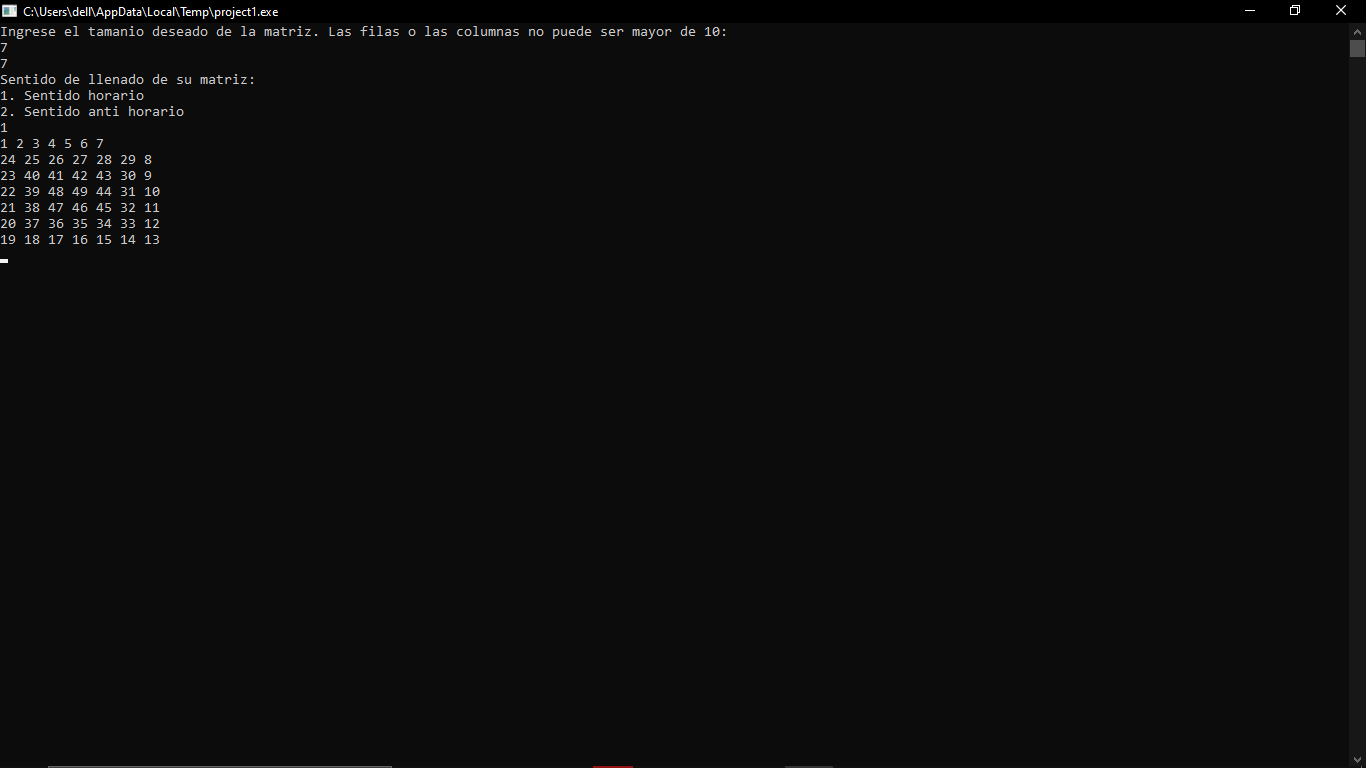
**Ejemplo 1:**



**Ejemplo 2:**



**Ejemplo 3:**



**EJERCICIO 2**

2. Dado N, un número impar, elaborar un algoritmo que genere un cuadrado mágico de orden N. El método de generación, de un cuadrado mágico, consiste en situar el número 1 en el centro de la primera fila; el número siguiente en la casilla situada por encima y a la derecha, y así sucesivamente. Debe considerarse, que el cuadrado es cíclico, lo que significa, que la línea encima de la primera fila es la última fila y la columna a la derecha de la última es la primera columna. Debe tomar en cuenta, que en el caso de que el número generado caiga en una casilla ocupada, se elige la casilla debajo del último número situado. A continuación, un ejemplo de un cuadrado mágico de orden 3. (5 ptos)

**ANALISIS:**

|  |  |  |
| --- | --- | --- |
| **Entradas** | **Procesos** | **Salida** |
| Número impar:  N | Pedir los datos de entrada.  Verificar si el número ingresado es impar.  Localizar la posición donde se pondrá el primer número.  Rellenar el cuadrado, según los parámetros dados.  Mostar los datos de salida. | Mostrar el cuadrado mágico relleno. |

**ALGORITMO EN PSEUDOCÓDIGO:**

**Cabecera** CuadradoMagico;

**Declaraciones**

i, j, N, Cont\_Num, Salvi, Salvj: Entero;

Matriz: Arreglo [1..9,1..9] Entero;

**Cuerpo**

**Inicio**

N <- 0;

Cont\_Num <- 1;

Repetir Para i, 1, N, +1, Hacer

Repetir Para j, 1, N, +1, Hacer

Matriz [i,j] := 0;

Fin-RP;

Fin-RP;

Repetir

Escribir (“Ingrese el tamaño deseado para su cuadrado mágico (La cantidad que ingrese no debe ser mayor a 10 y debe ser un número impar): “);

Leer (N);

Si (N>9) O (N Mod 2 = 0) Entonces

Escribir (“Ingreso un valor invalido. Vuelva a intentar.”);

Fin-Si;

Hasta (N<=9) Y (N Mod 2 <> 0);

Fin-RH;

i <- 1;

j <- N Div 2 + 1;

Salvi <- 0;

Salvj <- 0;

Repetir Mientras (Cont\_Num <=N\*N) Hacer

Si (i = 0) Entonces

i <- N;

Fin-Si;

Si (j > N) Entonces

j <- 1;

Fin-Si;

Si (Matriz [i,j] = 0) Entonces

Matriz[i,j] <- Cont\_Num;

Salvi <- i;

Salvj <- j;

Cont\_Num <- Cont\_Num + 1;

De lo contrario

Si (Matriz[i,j] <> 0) Entonces

i <- Salvi + 1;

j <- Salvj;

Matriz [i,j] <- Cont\_Num;

Salvi <- i;

Salvj <- j;

Cont\_Num <- Cont\_Num + 1;

Fin-Si;

Fin-Si;

i <- i - 1;

j <- j + 1;

Fin-RM;

Repetir Para i, 1, N, +1 Hacer

Repetir Para j, 1, N, +1 Hacer

Escribir (Matriz[i,j], “ “);

Fin-RP;

Escribir ();

Fin-RP;

**Fin.**

**PROGRAMA FUENTE:**

Program CuadradoMagico;

Uses Crt;

Var

i, j, N, Cont\_Num, Salvi, Salvj: Integer;

Matriz: Array [1..9,1..9] Of integer;

Begin

//Inicializando parte de las variables

N := 0;

Cont\_Num := 1;

//Limpieza de la matriz

For i := 1 To N Do

For j := 1 To N Do

Matriz [i,j] := 0;

//Validando los datos de entrada;

Repeat

Writeln ('Ingrese el tamanio deseado para su cuadrado magico (La cantidad que ingrese no debe ser mayor a 10 y debe ser un numero impar): ');

Readln (N);

If (N>9) Or (N Mod 2 = 0) Then

Writeln ('Ingreso un valor invalido. Vuelva a intentar.');

Until (N<=9) And (N Mod 2 <> 0);

//Inicializando las variables restantes

i := 1;

j := N Div 2 + 1;

Salvi := 0;

Salvj := 0;

While (Cont\_Num <=N\*N) Do

//Verificando si las variables excede los limites de la matriz

Begin

If (i = 0) Then

i := N;

If (j > N) Then

j := 1;

//Se comienza el llenado de la matriz

//Si en la casilla actual se encuentra un cero, entonces se escribira el numero correspondiente de la secuencia en dicha casilla

If (Matriz [i,j] = 0) Then

Begin

Matriz[i,j] := Cont\_Num;

//Se guarda la posicion en donde se encuentra el puntero actualmente

Salvi := i;

Salvj := j;

Cont\_Num := Cont\_Num + 1;

End

Else

//Si en la casilla actual se encuentra un numero diferente de cero, el numero correspondiente de la secuencia, se escribira debajo del ultimo numero escrito

If (Matriz[i,j] <> 0) Then

Begin

//Ubicamos el puntero en la casilla debajo del ultimo numero escrito

i := Salvi + 1;

j := Salvj;

Matriz [i,j] := Cont\_Num;

//Se guarda la posicion en donde se encuentra el puntero actualmente

Salvi := i;

Salvj := j;

Cont\_Num := Cont\_Num + 1;

End;

i := i - 1;

j := j + 1;

End;

//Se muestra la matriz rellena

For i := 1 To N Do

Begin

For j := 1 To N Do

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

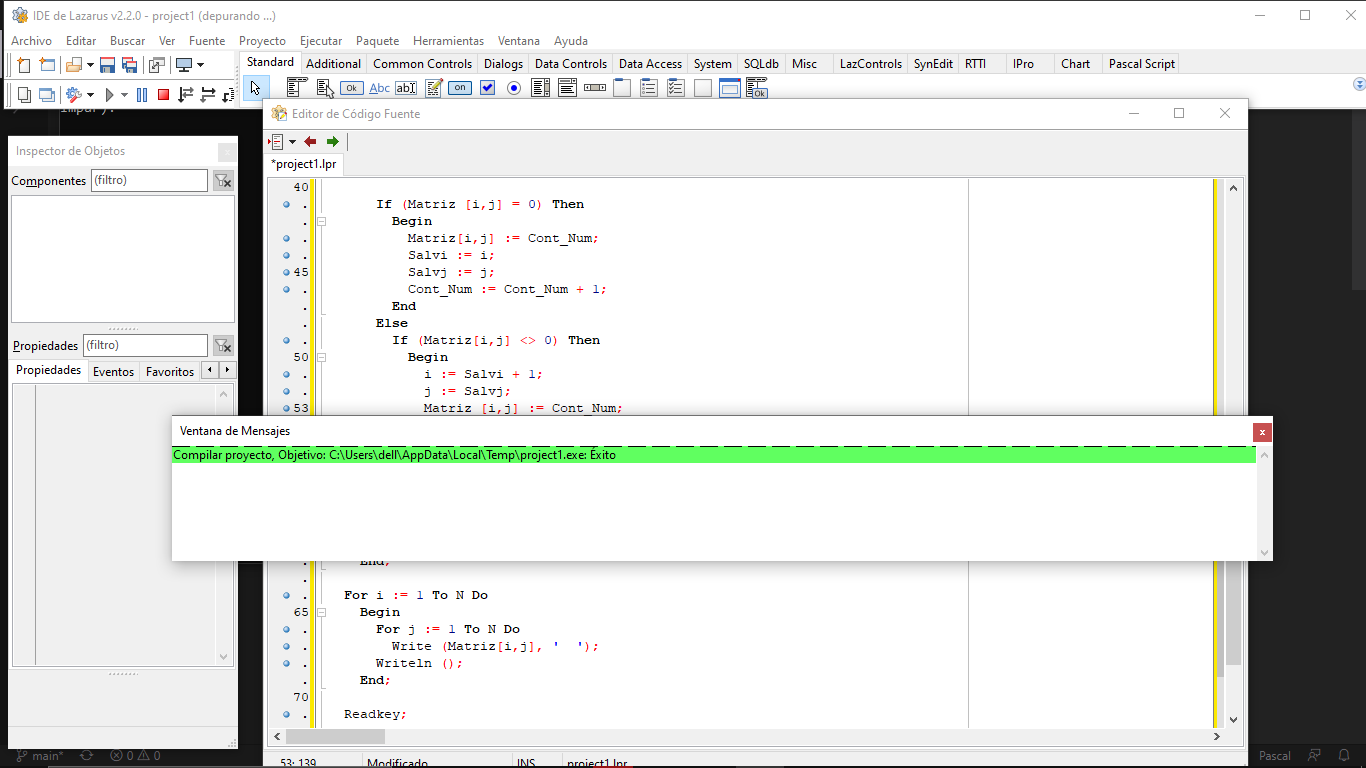
Writeln ();

End;

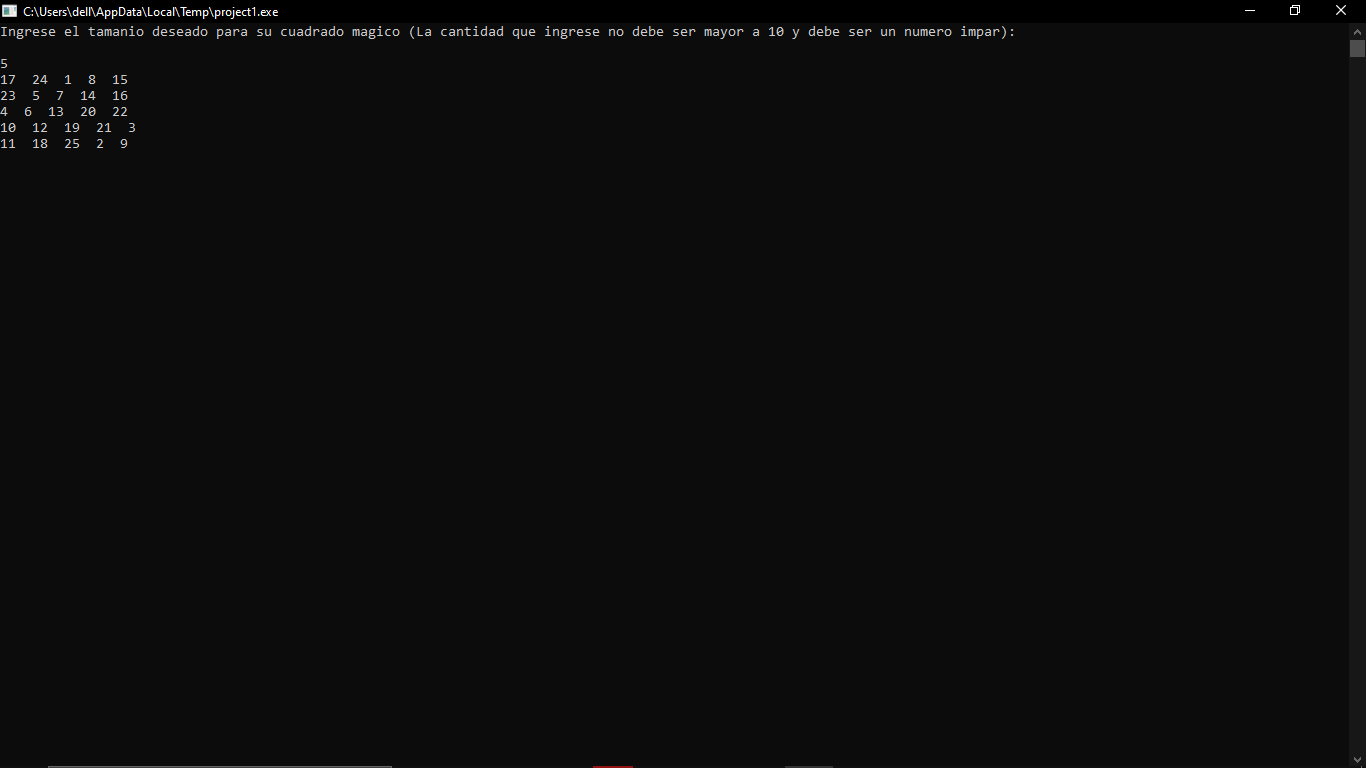
Readkey;

End.

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



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



**EJERCICIO 3**

**ANALISIS:**

|  |  |  |
| --- | --- | --- |
| **Entradas** | **Procesos** | **Salida** |
| Cantidad de hombres y mujeres contagiados: K | Limpiar las matrices.  Pedir los datos de entrada.  Recorrer la matriz tridimensional para calcular la cantidad de mujeres contagiadas en cada estado y guardarlo en una matriz unidimensional.  Recorrer la matriz tridimensional para calcular la cantidad de hombres contagiados durante un mes, dado por el usuario.  Recorrer la matriz tridimensional para calcular El estado con mayor número de contagios  Mostrar los datos de salida. | Mostrar Cantidad de mujeres contagiados en cada estado, cantidad de hombres contagiadas durante un mes dado y el estado con mayor número de contagio. |

**ALGORITMO EN PSEUDOCÓDIGO:**

**Cabecera:** CasosdeCovid;

**Declaraciones**

generos: Arreglo [1..2] de tipo cadena[7] = (“Hombres”, “Mujeres”);

Estados: Arreglo [1..23] de tipo cadena[16] = ('Amazonas', 'Anzoategui', 'Apure', 'Aragua', 'Barinas', 'Bolivar', 'Carabobo', 'Cojedes', 'Delta Amacuro', 'Falcon', 'Guarico', 'Lara',' Merida', 'Miranda', 'Monagas', 'Nueva Esparta', 'Portuguesa', 'Sucre', 'Tachira', 'Trujillo', 'Vargas', 'Yaracuy', 'Zulia');

meses: arreglo [1..12] de tipo cadena[10] = ('enero','febrero','marzo','abril','mayo', 'junio','julio','agosto','septiempre','octubre','noviembre','diciembre');

Casos: Arreglo [1..23,1..12,1..2] de tipo entero;

Mujeres\_Contg\_Est: Arreglo [1..23] de tipo entero;

i, j, k, mes, hombres\_Contg\_Mes, Acum\_estado\_Contg, Salv\_Acum\_estado\_Contg: Entero;

mayor: Cadena [16];

**Cuerpo**

**Inicio**

i <- 0;

j <- 0;

k <- 0;

Salv\_Acum\_estado\_Contg<- 0;

hombres\_Contg\_Mes<- 0;

Repetir Para i, 1, 23, +1, Hacer

Repetir Para j, 1, 12, Hacer

Repetir Para k, 1, 2, +1, Hacer

Casos[i,j,k]<- 0;

Fin-RP;

Fin-RP;

Fin-RP;

Repetir Para i, 1, 23, +1, Hacer

Mujeres\_Contg\_Est[i]<- 0;

Fin-RP;

Repetir Para i, 1, 23, +1, Hacer

Repetir Para, j, 1, 12, +1, Hacer

Repetir Para k, 1, 2, +1, Hacer

Escribir (“Ingrese la cantidad de contagiados de COVID-19 del Estado: “, Estados[i], “ ; del mes: “, Meses[j], “ ; de sexo: “, Generos[k]);

Leer (Casos [i,j,k]);

Fin-RP;

Fin-RP;

Fin-RP;

Repetir Para i, 1, 23, +1 Hacer

Repetir Para j, 1, 12, +1, Hacer

Mujeres\_Contg\_Est[i] <- Mujeres\_Contg\_Est[i] + Casos [i,j,2];

Fin-RP;

Fin-RP;

Escribir (“Indique el mes deseado, para saber la cantidad de hombres contagiados en dicho mes: “);

Repetir para i, 1, 12, +1, Hacer

Escribir (i,”.”, Meses[i]);

Fin-RP;

Leer (mes);

Repetir Para i, 1, 23, +1, Hacer

hombres\_Contg\_Mes := hombres\_Contg\_Mes + Casos [i, mes, 1];

Fin-RP;

Repetir Para i, 1, 23, +1, Hacer

Acum\_estado\_Contg <- 0;

Repetir Para j, 1, 12, Hacer

Repetir Para k, 1, 2, +1, Hacer

Acum\_estado\_Contg<- Acum\_estado\_Contg + Casos[i,j,k];

Fin-RP;

Fin-RP;

Si (Acum\_estado\_Contg > Salv\_Acum\_estado\_Contg) Entonces

Mayor <- Estados[i];

Salv\_Acum\_estado\_Contg<- Acum\_estado\_Contg;

Fin-Si;

Fin-RP;

Escribir (“Cantidad de mujeres contagiadas en cada estado:”);

Repetir Para i, 1, 23, +1, Hacer

Escribir (“Estado (“, Estados[i], “): “, Mujeres\_Contg\_Est[i]);

Fin-RP;

Escribir (“Cantidad de hombres contagiados durante el mes “, Meses[mes],”: “, hombres\_Contg\_Mes);

Escribir (“El estado con mayor numero de contagio es el: “, Mayor);

**Fin.**

**PROGRAMA FUENTE:**

Program CasosdeCovid;

Uses Crt;

Var

generos: Array [1..2] Of string[7] = ('Hombres', 'Mujeres');

Estados: array[1..23] Of string[16] = ('Amazonas', 'Anzoategui', 'Apure', 'Aragua', 'Barinas', 'Bolivar', 'Carabobo', 'Cojedes', 'Delta Amacuro', 'Falcon', 'Guarico', 'Lara',' Merida', 'Miranda', 'Monagas', 'Nueva Esparta', 'Portuguesa', 'Sucre', 'Tachira', 'Trujillo', 'Vargas', 'Yaracuy', 'Zulia');

meses: array[1..12] Of string[10] = ('enero','febrero','marzo','abril','mayo', 'junio','julio','agosto','septiempre','octubre','noviembre','diciembre');

Casos : Array [1..23,1..12,1..2] Of Integer;

Mujeres\_Contg\_Est: Array [1..23] Of integer;

i, j, k, mes, hombres\_Contg\_Mes, Acum\_estado\_Contg, Salv\_Acum\_estado\_Contg: Integer;

mayor: String [16]

Begin

//Inicializacion de algunas variables

i := 0;

j := 0;

k := 0;

Salv\_Acum\_estado\_Contg := 0;

hombres\_Contg\_Mes := 0;

//Limpieza de la matriz casos

For i:= 1 To 23 Do

For j := 1 To 12 Do

For k := 1 To 2 Do

Casos[i,j,k] := 0;

//Limpieza de la matriz Mujeres\_Contg\_Est

For i := 1 To 23 Do

Mujeres\_Contg\_Est[i] := 0;

For i:= 1 To 23 Do

Begin

Clrscr;

For j := 1 To 12 Do

For k := 1 To 2 Do

Begin

Writeln ('Ingrese la cantidad de contagiados de COVID-19 del Estado: '

, Estados[i], ' ; del mes: ', Meses[j], ' ; de sexo: ',

generos[k]);

Readln (Casos [i,j,k]);

End;

End;

//calculo de la cantidad de mujeres contagiadas en cada estado

clrscr;

For i := 1 To 23 Do

For j := 1 To 12 Do

Mujeres\_Contg\_Est[i] := Mujeres\_Contg\_Est[i] + Casos [i,j,2];

//Cantidad de hombres contagiados durante un mes dado

Writeln ('Indique el mes deseado, para saber la cantidad de hombres contagiados en dicho mes: ');

For i:= 1 To 12 Do

Writeln (i,'.', Meses[i]);

Readln (mes);

For i := 1 To 23 Do

hombres\_Contg\_Mes := hombres\_Contg\_Mes + Casos [i, mes, 1];

//Calculo del estado con mayor numero de contagio

For i := 1 To 23 Do

Begin

Acum\_estado\_Contg := 0;

For j := 1 To 12 Do

For k := 1 To 2 Do

Acum\_estado\_Contg := Acum\_estado\_Contg + Casos[i,j,k];

If (Acum\_estado\_Contg > Salv\_Acum\_estado\_Contg) Then

Begin

Mayor := i;

Salv\_Acum\_estado\_Contg := Acum\_estado\_Contg;

End;

End;

//Salida

Writeln ('Cantidad de mujeres contagiadas en cada estado:');

For i := 1 To 23 Do

Writeln ('Estado (',Estados[i],'): ', Mujeres\_Contg\_Est[i]);

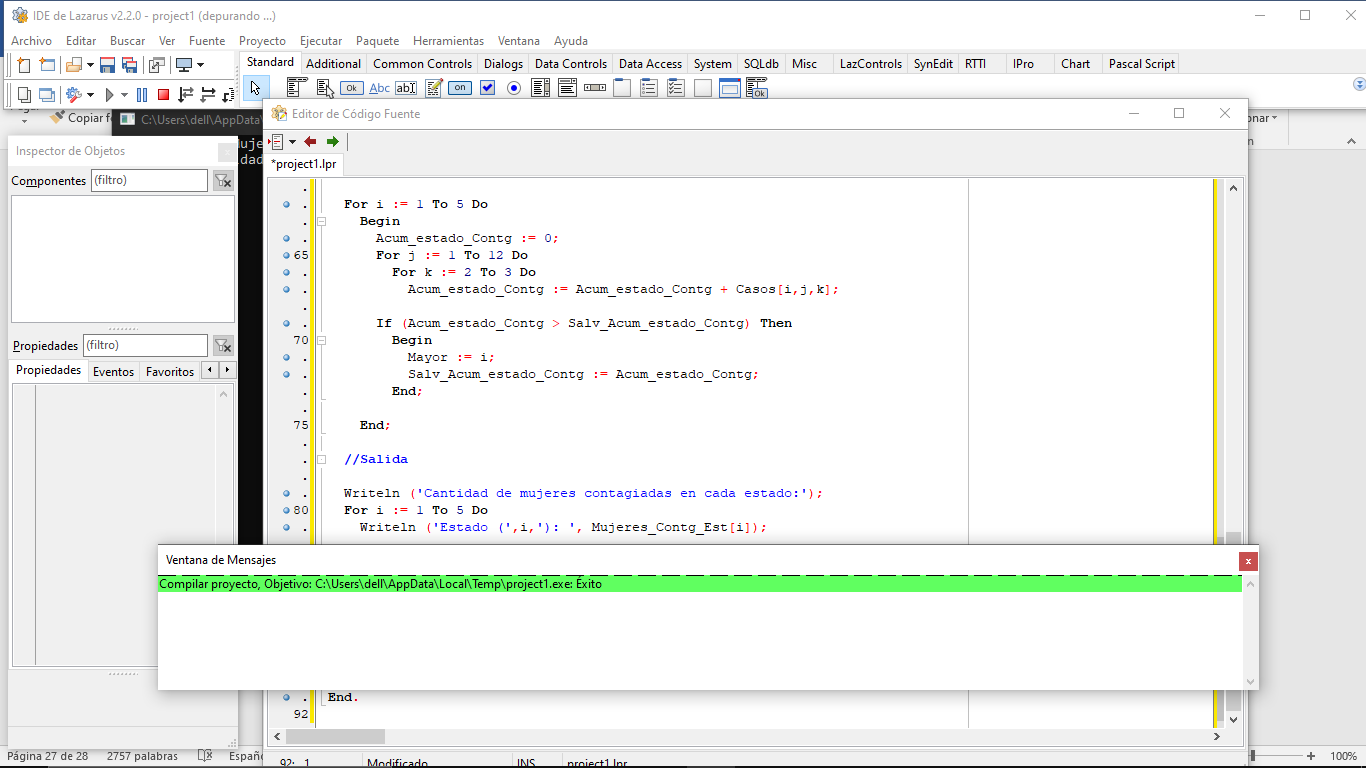
Writeln ('Cantidad de hombres contagiados durante el mes ', Meses[mes],': ', hombres\_Contg\_Mes);

Writeln ('El estado con mayor numero de contagio es el: ', Mayor) ;

Readkey;

End.

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



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

