

## Capítulo 4 : ESTRUCTURA DE REPETICION

1. Ciclo Incondicional (For)
2. Ciclo Condicional (While)
3. Ciclo Condicional (Repeat)
4. Ejemplos (While – Repeat - For)
5. Anidación de ciclos (While – Repeat - For).

### Estructura de Control de Repetición, Iterativa o Ciclos

Se utilizan para ejecutar sentencias en forma repetitiva. Si el número de repeticiones se conoce de antemano (o sea antes que las repeticiones comiencen) se utiliza la sentencia For, While y Repeat.

#### 1 Ciclo Incondicional

La característica de este ciclo es que se ejecuta una cantidad “conocida” de veces, la propia estructura asume el control a través de la variable de control de ciclo (VC) que toma valores ordinales progresivos (o regresivos), determinados entre un límite inicial y otro final (VI y VF son expresiones ordinales).

Según sea la secuencia ascendente o descendente

##### 1.1 FOR ascendente (For – to ; VI<=VF )

```
For VC := VI to VF do  
    Sentencia;
```

VC: variable de control. VI: valor inicial VF: valor final Sentencia simple o compuesta
--------------------------------------------------------------------------------------------------

La variable VC toma el valor VI y se incrementa en 1 en cada iteración hasta alcanzar el valor VF,

Previo a cada iteración se verifica que se cumpla  $VC \leq VF$ . Estas acciones las realiza la estructura en forma automática.

##### 1.2 FOR descendente (For-downto ; VI>=VF)

```
For VC := VI downto VF do  
    Sentencia;
```

El cambio que conlleva con respecto al anterior es que en cada iteración decrementa en 1 la variable de control (VC)

Previo a cada iteración se verifica que se cumpla  $VC \geq VF$ .

#### Consideraciones:

- El valor inicial y el valor final se evalúan al comienzo (una sola vez) y si el valor inicial es mayor (o menor en el downto) que el final no se ejecuta el ciclo.
- La variable de control de ciclo (VC) y los límites inicial y final (VI y VF) NO deben ser alterados dentro del ciclo For.
- La variable de control queda indefinida luego de terminar For.
- Es posible calcular la cantidad de repeticiones a partir de VI, VF y el Paso (1 ó -1).  
 $\text{CantRepeticiones} = (VF - VI) * \text{Paso} + 1$

Ejemplo1 - Leer un conjunto de números enteros (distintos de cero),  
a – son 5 números, mostrar los impares.  
b – son N números.

<pre> Program Ej1a; Var i: word; Num: integer; Begin <b>For i:= 1 to 5 do</b>   Begin     Writeln('Ingresar un numero:');     Readln(Num);     If Num MOD 2&lt;&gt; 0 then       Writeln(Num);     End   End. </pre>	<pre> Program Ej1b; Var i, N: word; Num: integer; Begin <b>Write ('Ingresar la cantidad de numeros:');</b> <b>Readln(N);</b> <b>For i:= 1 to N do</b>   Begin     Writeln('Ingresar un numero:');     Readln(Num);     If Num MOD 2&lt;&gt; 0 then       Writeln(Num);     End   End. </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Es común que algunas de las sentencias que se ejecutan dentro del ciclo cuenten ocurrencias (de las repeticiones del ciclo o de determinados eventos) o acumulen sumas (o productos).

- Un contador es una variable entera que se incrementa (decrementa) en uno.  
Cont:= Cont + 1;
- Un acumulador es una variable numérica que se incrementa (decrementa) en una cantidad variable.  
Acum:=Acum + X;

Ejemplo2 - Leer un conjunto de N números enteros (distintos de cero),  
a - mostrar los impares y además contar e informar cuantos son pares.  
b - calcular el promedio de positivos y negativos

```

Program Ej2a;
Var
i, ContP, N: word;
Num: integer;
Begin
ContP:=0;
Write ('Ingresar la cantidad de numeros:');
Readln(N);
For i:= 1 to N do
  Begin
    Writeln('Ingresar un numero:');
    Readln(Num);
    If Num MOD 2<> 0 then
      Writeln(num)
    Else
      ContP:= ContP +1;
    End;
  Writeln('La cantidad de pares es ', ContP);
End.

```

```
Program Ej2b;
Var
  N, i, ContP, ContN, SumP: word;
  Num, SumN: integer;
Begin
  ContP:=0; ContN:=0;
  SumP:=0; , SumN:=0;
  Write ('Ingresar cantidad de numeros:');
  Readln(N);
  For i:= 1 to N do
    Begin
      Writeln('Ingresar un numero:');
      Readln(Num);
      If Num > 0 then
        Begin
          ContP:= ContP +1;
          SumP:= SumP + Num;
        End
      Else
        Begin
          ContN:= ContN +1;
          SumN:= SumN + Num;
        End
      End;
  If ContP <> 0 then
    Writeln('Promedio positivos', SumP DIV ContP)
  Else
    Writeln('no ingresaron numeros positivos');
  If ContN <> 0 then
    Writeln('Promedio negativos', SumN DIV ContN)
  Else
    Writeln('no ingresaron numeros negativos');
End.
```



Resolver los ítems 1 a 3 de los ejercicios propuestos al final del capítulo.

Ejemplo3 - Leer un conjunto de N números reales calcular e informar el máximo  
a – iniciando el máximo con un valor imposible  
b – iniciando el máximo con el 1er valor leído

<pre>{inicia el máximo con un valor imposible} Program Ej3a; Var   N, i: byte;   Num, Max: real; Begin Write ('Ingresar cantidad de numeros:'); Readln(N); <b>Max:= -9999;</b> For i:= 1 to N do   Begin     Write('Ingresar un numero:');     Readln(num);     If Num &gt; Max then       Max := Num;     End   Writeln('El maximo es', Max); End.</pre>	<pre>{inicia el máximo con el primer valor leído} Program Ej3b; Var   N, i: byte;   Num, Max: real; Begin Write ('Ingresar cantidad de numeros:'); Readln(N); <b>Write ('Ingresar un numero:');</b> <b>Readln(Num);</b> <b>Max:= Num;</b> For i:= 2 to N do   Begin     Write('Ingresar un numero:');     Readln(Num);     If Num &gt; Max then       Max := Num;     End   Writeln('El maximo es', Max); End.</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Notar que las dos soluciones anteriores, difieren en la forma en la que la variable Max toma su primer valor. Si bien la segunda es más eficiente, no siempre es posible implementarla. Por ejemplo si se pide el máximo de los pares para los 6 siguientes datos:  
 91, 8, 45, 17, 20, 54  
 iniciar con el primer valor no permitiría seleccionar 54 como máximo.



Resolver el ítem 4 de los ejercicios propuestos al final del capítulo.

## Ciclo Condicional

La característica de estos ciclos es que no se conoce la cantidad de repeticiones, están “condicionados” por una expresión lógica cuyo resultado determina la continuación o la finalización del ciclo.

Generalmente la expresión lógica utiliza operadores relacionales para enfrentar un valor preestablecido con una variable, que se actualiza por lectura o cálculo.

Estas situaciones pueden combinarse mediante el uso de operadores lógicos, que potencian el control que se ejerce sobre el ciclo.

### 2. Ciclo While

While Expresión lógica do Sentencia;	While Expresión lógica do Begin Sentencia <sub>1</sub> ; Sentencia <sub>2</sub> ; ..... Sentencia <sub>n</sub> ; End;
-----------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------

Se evalúa la expresión lógica, si es verdadera ejecuta la sentencia (simple o compuesta) y vuelve a evaluar la expresión. Si es falsa el ciclo finaliza y continua la ejecución de la sentencia que está físicamente abajo del ciclo.

#### Notar que:

- Evalúa la expresión lógica al comienzo, por lo tanto, si la primera vez resulta falsa no entra al ciclo y no se ejecuta ni una vez.
- Algunas de las variables que forman parte de la expresión lógica deben modificarse (por lectura o cálculo) durante la ejecución del ciclo, para que la expresión pase de resultar verdadera a resultar falsa y así detener el ciclo. De otra forma el ciclo sería **infinito**.

**2.1.** La condición (expresión lógica) que controla el ciclo While verifica el valor (“imposible” o no) ingresado por lectura, deteniendo o continuando el ciclo.

Ejemplo4 - Leer un conjunto de números enteros cuya cantidad no se conoce de antemano (se sabe que son distintos de cero),

a - mostrar los impares

b - mostrar los impares y además contar e informar cuantos son pares.

<pre> Program Ej4a; Var   Num: integer; Begin   Write ('Ingresar un numero:');   Readln(Num);   While Num &lt;&gt; 0 do     Begin       If Num MOD 2&lt;&gt; 0 then         Writeln(num);       Write('Ingresar un numero:');       Readln(Num);     End   End. </pre>	<pre> Program Ej4b; Var   Num: integer;   ContP: word; Begin   ContP:=0;   Write ('Ingresar un numero:');   Readln(Num);   While Num &lt;&gt; 0 do     Begin       If Num MOD 2&lt;&gt; 0 then         Writeln(num)       Else         ContP:= ContP +1;       Write('Ingresar un numero:');       Readln(Num);     End;   Writeln('La cantidad de pares es ', ContP); End. </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

c - calcular el máximo (iniciando el máximo con un valor imposible o con el 1er valor leído)

<pre> {inicia el máximo con un valor imposible} Program Ej4c1; Var   Num, Max: integer; Begin   Max:= -9999;   Write ('Ingresar un numero:');   Readln(Num);   While Num &lt;&gt; 0 do     Begin       If Num &gt; Max then         Max := Num;       Write('Ingresar un numero:');       Readln(num);     End   Writeln('El maximo es', Max); End. </pre>	<pre> {inicia el máximo con el 1er valor leído} Program Ej4c2; Var   Num, Max: integer; Begin   Write ('Ingresar un numero:');   Readln(Num);   Max:= Num;   While Num &lt;&gt; 0 do     Begin       If Num &gt; Max then         Max := Num;       Write('Ingresar un numero:');       Readln(num);     End   Writeln('El maximo es', Max); End. </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- d - calcular el promedio de positivos y negativos.  
 e - calcular el promedio de positivos, negativos y contar los ceros

```

Program Ej4d;
Var
  ContP, ContN, SumP: word;
  Num, SumN: integer;
Begin
  ContP:=0; ContN:=0;
  SumP:=0; SumN:=0;
  Write ('Ingresar un numero:');
  Readln(Num);
  While Num <> 0 do
    Begin
      If Num > 0 then
        Begin
          ContP:= ContP +1;
          SumP:= SumP + Num;
        End
      Else
        Begin
          ContN:= ContN +1;
          SumN:= SumN + Num;
        End
      Write('Ingresar un numero:');
      Readln(num);
    End;
  If ContP <> 0 then
    Writeln('Promedio positivos',
      SumP DIV ContP)
  Else
    Writeln('no ingresaron numeros positivos');
  If ContN <> 0 then
    Writeln('Promedio negativos',
      SumN DIV ContN)
  Else
    Writeln('no ingresaron numeros
negativos');
End.

```

```

Program Ej4e;
Var
  ContP, ContN, SumP, ContC: word;
  Num, SumN: integer;
  Res : char;
Begin
  ContP:=0; ContN:=0; ContC:=0;
  SumP:=0; SumN:=0;
  Write ('Desea ingresar un numero?(S/N): ');
  Readln(Res);
  While Res <> 'S' do
    Begin
      Readln(Num);
      If Num > 0 then
        Begin
          ContP:= ContP +1;
          SumP:= SumP + Num;
        End
      Else
        If Num < 0 then
          Begin
            ContN:= ContN +1;
            SumN:= SumN + Num;
          End
        Else
          ContC:= ContC +1;
      Write('Ingresa otro numero?(S/N):');
      Readln(Res);
    End;
  If ContP <> 0 then
    Writeln('Promedio positivos',
      SumP DIV ContP)
  Else
    Writeln('no ingresaron numeros positivos');
  If ContN <> 0 then
    Writeln('Promedio negativos',
      SumN DIV ContN)
  Else
    Writeln('no ingresaron numeros negativos');
  Writeln('ingresaron ',ContC,' ceros')
End.

```



Resolver los ítems 5 y 6 de los ejercicios propuestos al final del capítulo.

**2.2.** La condición que controla el ciclo While verifica si el resultado de un cálculo ha alcanzado o no un valor límite (ingresado por lectura), deteniendo o continuando el ciclo.

Las condiciones de los ciclos condicionales pueden ser compuestas (operadores lógicos)

Ejemplo5 – Leer un conjunto de números reales y sumarlos, detener el proceso cuando

a - la suma supere un cierto valor X, ingresado por teclado. Mostrar la cantidad de números sumados y la suma obtenida.

b - la suma supere un cierto valor X, o la cantidad de número leídos sea mayor a M (ambos valores ingresados por teclado). Mostrar la cantidad de números sumados y la suma obtenida.

```
Program Ej5a;
Var
  Num, X, Sum: real;
  Cont: word;
Begin
  Sum:=0;
  Cont:= 0;
  Writeln('Ingresar la cota de la suma:');
  Readln(X);
  While Sum < X do
    Begin
      Writeln('Ingresar un numero:');
      Readln(Num);
      Sum:= Sum + Num;
      Cont:= Cont +1;
    End;
  Writeln('Son', Cont, 'numeros');
  Writeln('Su suma es', Sum:8:2);
End.
```

```
Program Ej5b;
Var
  Num, X, Sum: real;
  M, Cont: word;
Begin
  Sum:=0;
  Cont:= 0;
  Writeln('Ingresar la cota de la suma y de los numeros:');
  Readln(X, M);
  While (Sum < X) and (Cont < M) do
    Begin
      Writeln('Ingresar un numero:');
      Readln(Num);
      Sum:= Sum + Num;
      Cont:= Cont +1;
    End;
  Writeln('Son', Cont, 'numeros');
  Writeln('Su suma es', Sum:8:2);
End.
```

Es importante que alguna de las variables que intervienen en la expresión lógica que controla el ciclo While, cambien su valor dentro del ciclo, en caso contrario el ciclo es infinito (no se detiene). Si las siguientes sentencias faltaran en los ejemplos anteriores,

Readln(num);

Sum:= Sum + Num;

Cont:= Cont +1;

se generan ciclos infinitos.



Resolver el ítem 7 de los ejercicios propuestos al final del capítulo.



### 2.3. Ciclo while para control de lectura de datos desde archivo

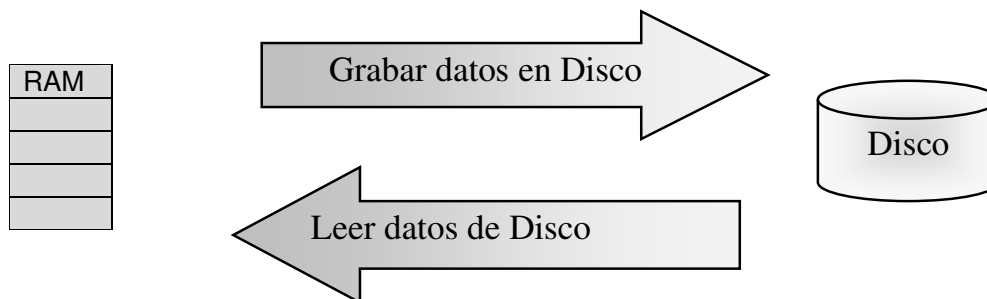
Un caso particular es el ciclo que lee datos de un archivo, la función booleana `eof()` que controla si se alcanzó o no el fin de archivo constituye la expresión lógica que controla el fin o la continuación del ciclo, respectivamente.

Tanto la entrada como la salida estándar, usan los archivos del sistema **input** y **output** por defecto, o sea que no es necesario especificar el teclado y la pantalla respectivamente.

Si se requiere lectura o escritura sobre archivo es necesario indicar el nombre del archivo (y en el caso de que este no se encuentre en la misma ubicación del programa, también indicar unidad y carpeta que lo contiene), como así también la operación a realizar sobre el mismo (leer o escribir).

#### Consideraciones previas para lectura o escritura desde un almacenamiento secundario con archivos de texto (*text*)

Si el programador desea ingresar datos desde el disco rígido o removible (donde fueron almacenados previamente como un **archivo de texto**), debe indicar al programa el nombre del archivo que contiene los datos y su intención de leer datos desde ese archivo. De manera similar, puede crearse y almacenar datos en un archivo de texto. Los archivos de texto en Pascal requieren de variables de tipo **text**. Estos archivos son secuenciales, es decir se acceden en forma progresiva desde el comienzo al final, esta característica los hace lentos y se recomienda su uso con información que será procesada en su totalidad.



El programa Pascal se ejecuta desde la memoria de acceso aleatorio (o RAM). Si el programador desea almacenar datos en un archivo de texto en el disco, estos deben ser grabados con sentencias específicas de escritura. Por otro lado, si desea leer o cargar datos ya previamente almacenados en el disco para llevarlos a la memoria, deben ser recuperados con otras sentencias específicas. Un archivo de texto es utilizado para leer o grabar, en forma excluyente o sea no admite ambas operaciones simultáneamente.

Los datos en un archivo de texto se almacenan usando el código ASCII, en el cual cada carácter es representado por un simple byte. Debido a que los archivos de texto utilizan el código ASCII, se pueden ver o imprimir. En este tipo de archivos, todos sus datos se almacenan como cadenas de caracteres, es decir, los números se almacenan con su representación ASCII y no su representación numérica. Esto implica una conversión automática en el almacenamiento y la recuperación, donde un valor numérico convertido a cadena, vuelve a su representación original (cuando el destino es una variable numérica).

Luego de declarar la variable de tipo archivo en la sección

Var

NombreArch:text;

Deben utilizarse las siguientes sentencias, de cada uno se describe la sintaxis y su función:

Sentencia	Sintaxis	Función
Assign	assign(nombre archivo Pascal, nombre archivo en el disco)	Enlaza la variable de tipo text con el archivo físico en el disco. Necesario para lectura/ escritura.
Reset	reset(nombre del archivo Pascal)	Prepara al archivo para la lectura, ubicándose al comienzo.
read/ readln	read/readln(nombre archivo Pascal, lista de variables)	Se leen del archivo los valores y se almacenan en las variables indicadas en la lista.
Rewrite	rewrite(nombre del archivo Pascal)	Prepara al archivo para la escritura, ubicándose al comienzo.
write/ writeln	write/writeln(nombre del archivo Pascal, lista de variables )	Se graban en el archivo los valores contenidos en las variables de la lista.
Close	close(nombre del archivo Pascal)	Cierra el archivo.
Eof	eof(nombre de archivo Pascal)	Función booleana que detecta el fin del archivo

Ha de tenerse cuidado con la sentencia **rewrite**, pues si el archivo no existe, se crea uno nuevo, pero si ya existe, *su contenido se pierde al sobrescribir los nuevos valores*.

Ejemplo6 - se reformula el ejemplo 4b) considerando que los números enteros ingresan desde un archivo Numeros.txt (no se sabe cuántos son), están grabados uno por línea. Calcular la cantidad de números pares y mostrar los impares.

Program Ej6;

Var

Num: integer;

ContP: word;

**Arch: text;** {se declara una variable para acceder al archivo de texto}

Begin

ContP:=0;

**Assign (Arch,'Numeros.txt');**{vincula el archivo Numeros.txt con la variable Arch}

**Reset (Arch);** {prepara el archivo para lectura}

**While not Eof(Arch) do** {controla que no haya alcanzado el fin de archivo}

Begin

**Readln(Arch, Num);** {lee un número en una línea del archivo y salta a la siguiente}

If Num MOD 2<> 0 then

Writeln(Num)

Else

ContP:= ContP +1;

End;

Writeln('La cantidad de pares es ', ContP);

**Close (Arch);** {cierra el archivo}

End.

Ejemplo7 - En el archivo Clima.txt se ha grabado información climática, en cada línea:  
Fecha (dd/mm/aaaa) máxima mínima

Se pide leer el archivo y grabar en otro archivo Amplitud.txt las fechas que registren amplitud térmica mayor a 10.

```
program Ej7;
var
  fecha:string[10];
  max, min:real;
  ArchE, ArchS:text; {se declaran dos archivos, se usara uno de entrada y otro de salida}
Begin
  Assign (ArchE,'Clima.txt'); Reset (Arch);      {prepara el archivo para lectura}
  Assign (ArchS,'Amplitud.txt'); Rewrite(ArchS); {prepara el archivo para escritura}
  While not eof(ArchE) do
    Begin
      Readln(ArchE, fecha, max, min);             {lee en una línea una medición }
      If max - min > 10 then
        Writeln(ArchS, fecha);                    {si la amplitud es mayor a 10 graba la fecha}
      End;
    Close (ArchE); Close (ArchS);                 {cierra los archivos}
  End.
```



Resolver el ítem 8 de los ejercicios propuestos al final del capítulo.

### 3. Ciclo Repeat

#### REPEAT

Sentencia/s;

UNTIL expresión lógica;

Se ejecutan las sentencias que contiene la estructura y luego evalúa la condición, si es falsa repite el ciclo hasta que la condición sea verdadera, cuando el ciclo se detiene continúa la ejecución de la sentencia que esta físicamente abajo del mismo.

#### Notar que:

- Evalúa la condición después de ejecutar las sentencias que contiene, por lo tanto se ejecuta al menos una vez.
- Algunas de las variables que forman parte de la expresión lógica deben modificarse durante la ejecución del ciclo, para que la expresión pase de resultar falsa a resultar verdadera y así detener el ciclo.
- A diferencia del ciclo While que itera mientras la condición es verdadera, el Repeat itera si la condición resulta falsa (o sea se detiene cuando es verdadera).
- Para codificar una estructura While como una estructura Repeat, basta con negar la expresión lógica. Pero no es totalmente equivalente ya que la primera puede no ejecutarse nunca y la segunda se ejecuta al menos una vez.
- Las consideraciones hechas para la estructura While sobre las expresiones lógicas que controlan las repeticiones se aplican a la estructura Repeat (valor imposible por lectura, valor límite por cálculo, contador alcanza una cantidad establecida)

Se reprograman los ejemplos vistos anteriormente.

Ejemplo8 – Se reformula el ejemplo 4a) Leer un conjunto de números enteros cuya cantidad no se conoce de antemano (se sabe que son distintos de cero), mostrar los impares.

Ejemplo9 - Leer un mes (1 .. 12) luego indicar a que trimestre pertenece (1 .. 4) . Validar la lectura hasta que ingrese un mes correcto.

```
Program Ej8;
Var
  Num: word;
Begin
  Write ('Ingresar un numero:');
  Readln(Num);
  Repeat
    If Num MOD 2<> 0 then
      Writeln(num);
    Writeln('Ingresar un numero:');
    Readln(Num);
  Until Num = 0;
End.
```

```
Program Ej9;
var
  Mes:byte;
Begin
  Repeat
    writeln('ingrese mes');
    readln(Mes);
  Until (1<=Mes) and (Mes<=12);
  Writeln ('pertenece al trimestre ', (Mes-1) div 3
+1)
End.
```

Ejemplo10 - Leer dos números reales, y luego utilizar en forma repetitiva un **menú** para obtener uno o más de los siguientes resultados: suma, resta, producto o división.

Program Ej10;

var

x, y: real;

Op: char;

Begin

writeln('ingrese dos números distintos de cero'); readln(x, y);

Repeat

writeln('S - Suma');

writeln('R - Resta');

writeln('P - Producto');

writeln('D - Division');

writeln('F - Finalizar');

writeln('Ingrese opcion');

Readln(Op);

If Op = 'S' then

Writeln( x + y :8:2)

else

If Op= 'R' then

Writeln( x - y :8:2)

else

if Op = 'P' then

Writeln( x \* y :8:2);

else

If Op ='D' then

Writeln ( x / y :8:2);

Until Op = 'F';

End.

**menú**

*{otra posibilidad}*

Case Op of

'S' : writeln( x + y :8:2);

'R' : writeln( x - y :8:2);

'P' : writeln( x \* y :8:2);

'D' : writeln ( x / y :8:2);

else





writeln ( 'Opcion erronea')

end;

Los dos últimos ejemplos muestran las aplicaciones más comunes de la estructura Repeat.

#### 4. Ejemplos (While – Repeat - For)

Ejemplo11 - La admisión a la facultad depende del resultado obtenido en los exámenes de tres materias. Se tienen las notas (0-100) correspondientes a N aspirantes a ingresar a la universidad, por cada uno :

-  Nombre
-  Nota Matemáticas
-  Nota Física
-  Nota Química

Ingresa el alumno que haya cumplido uno de estos tres requisitos :

C1- Las tres notas mayores o iguales a 40

C2- Matemáticas  $\geq 60$  y Química  $\geq 80$

C3- Física + Química  $\geq 180$

Desarrollar una solución que ingrese los N datos, liste los nombres de los ingresantes y la condición que cumplen, e informe el % de los mismos.

*Se deben ingresar exactamente N datos, para ello se usa el ciclo For. En cada iteración se ingresan los datos de un aspirante, (nombre y 3 notas), y se analizan para determinar si cumple alguno de los tres requisitos (C1, C2 o C3). Considerar como precondition  $N > 0$ .*

Program Ej11;

Var

I, N, NM, NF, NQ, ContIng : byte;

Nomb: string[25];

begin

Readln(N);

ContIng:= 0;

For I := 1 to N do

begin

Readln(Nomb);

Readln(NM, NF, NQ);

If (NM  $\geq 40$ ) AND (NF  $\geq 40$ ) AND (NQ  $\geq 40$ ) then

Begin

writeln (Nomb, 'C1');

ContIng:= ContIng + 1;

end

else

if (NM  $\geq 60$ ) AND (NQ  $\geq 80$ ) then

Begin

writeln (Nomb, 'C2');

ContIng:= ContIng + 1;

end

else

if NF + NQ  $\geq 180$  then

Begin

writeln (Nomb, 'C3');

ContIng:= ContIng + 1;

End

End;

writeln ('El % de ingresantes es ', ContIng \* 100/ N: 8:2);

end.

¿Sería posible implementar CASE para controlar las diferentes alternativas?

Ejemplo12 - En un estudio de mercado (para el posicionamiento de un producto), se encuestó una cantidad no conocida de personas. Por cada una se obtuvo una de estas tres posibles respuestas:

N - No lo conoce

C - Lo conoce y lo consume

A - Lo conoce y no lo consume

Se pide desarrollar una solución que ingrese las respuestas (F es fin de datos), calcule e informe:


- ✓ % de personas que no lo consumen, sobre el total de encuestados
- ✓ % de personas que lo consumen, sobre el total de personas que lo conocen
- ✓ % de personas que no lo conocen, sobre el total de encuestados

Ejemplo:      C   C   N   C   N   A   C   N   F

ContN= 3      NoConsumen      = (3 + 1)\*100/8

ContC= 4      ConsumenConocen      = 4\*100/(4 + 1)

ContA= 1      NoConocen      = 3\*100/8



Program Ej12;

Var

ContT, ContC, ContN, ContA : word;

Res : char;

Begin

ContC:= 0;

ContN:= 0;

ContA:= 0;

Writeln('C- lo conoce y consume ; N – no lo conoce ; A - lo conoce y no lo consume; F - Fin');

Readln(Res);

Res:= UPCASE (Res);

While Res <> 'F' do

begin

If Res = 'C' then

ContC:= ContC + 1

Else

If Res = 'N' then

ContN:= ContN + 1

Else

ContA:= ContA + 1;

Writeln('C- lo conoce y consume ; N – no lo conoce ; A - lo conoce y no lo consume; F – fin');

Readln(Res);

Res:= UPCASE (Res);

End;

ContT:= ContC + ContN + ContA;

writeln('% de personas que no lo consumen', (ContN + ContA)/ContT \* 100 :8:2);

writeln ('% de personas que lo consumen', ContC/(ContC + ContA) \* 100 :8:2); (\*)

writeln ('% de personas que no lo conocen', ContN /ContT \* 100 :8:2);

end.

*{otra posibilidad}*

Case Res of

'C' : ContC:= ContC + 1;

'N' : ContN:= ContN + 1;

'A' : ContA:= ContA + 1;

End;

(\*) Si la encuesta solo registra datos de personas que no lo conocen, esta sería una división por cero. Por lo tanto debería controlarse esta situación utilizando una estructura alternativa (If).

Reprogramar ingresando las respuestas de la encuesta desde archivo

Ejemplo13 - Ingresar una secuencia de N números reales, verificar si esta ordenada en forma ascendente.

- Analizar los N números
- Solo analizar los números necesarios para determinar si están o no ordenados.

<p>a) Analizar los N números</p> <pre> ..... Readln (N); Readln( Pri); Asc := true; For i := 2 to N do begin   Readln(Seg);   If Pri &gt; Seg then     Asc := false;   Pri := Seg; End; If Asc then   writeln( 'La secuencia está ordenada') else   writeln( 'La secuencia No está ordenada'); End.</pre>	<p>b) Solo analizar los números necesarios para determinar si están o no ordenados.</p> <pre> ..... Readln (N); Readln( Pri); Asc := true; i := 2; While Asc and (i &lt;= N) do begin   Readln(Seg);   If Pri &gt; Seg then     Asc := false; Else   begin     Pri := Seg;     i:=i +1;   End; If Asc then   writeln( 'La secuencia está ordenada') else   writeln( 'La secuencia No está ordenada'); End.</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

La variable booleana Asc puede omitirse, reemplazándola por la relación (Pri < Seg) que determina su valor (True- False). Es necesario ingresar previo al ciclo los valores de Pri y Seg.

```

.....
Readln (N);
Readln( Pri, Seg);
i := 3;
While (i <= N) and (Pri < Seg) do
begin
  Pri := Seg;
  Readln(Seg);
  i:=i +1;
End;
If i > N then
  writeln( 'La secuencia está ordenada')
else
  writeln( 'La secuencia No está ordenada');
End.
```

Probar para los siguientes juegos de datos:

N= 5,      2, 7, 34, 65, 84  
 N= 7,      2, 7, 74, 65, 84, 89, 93





Resolver los ítems 9 y 10 de los ejercicios propuestos al final del capítulo.

## 5. Ciclos Anidados

### Ciclos Secuenciales

Son independientes, después que finaliza el primer ciclo, comienza el segundo.

Ejemplo:

```
Program CiclosSecuenciales;
```

```
Var
```

```
    Letra: Char;
```

```
    Nro:byte;
```

```
Begin
```

```
For Nro:= 1 to 5 do
```

```
    Write (Nro:3);
```

```
Writeln;
```

```
For Letra:= 'a' to 'z' do
```

```
    Write (letra:3);
```

```
End.
```

1 2 3 4 5

a b c d e f g h i j k l m n o p q r s t u v w x y z

Total de repeticiones 5 + 26

### Ciclos Anidados

Son dependientes, por cada iteración del ciclo externo se ejecuta íntegramente el ciclo interno.

Ejemplo:

```
Program CiclosAnidados;
```

```
Var
```

```
    Letra: Char;
```

```
    Nro:byte;
```

```
Begin
```

```
For Nro:= 1 to 5 do
```

```
    begin
```

```
        Write (Nro:3);
```

```
        For Letra:= 'a' to 'z' do
```

```
            Write (letra:3);
```

```
        Writeln;
```

```
    End;
```

```
End.
```

1 a b c d e f g h i j k l m n o p q r s t u v w x y z

2 a b c d e f g h i j k l m n o p q r s t u v w x y z

3 a b c d e f g h i j k l m n o p q r s t u v w x y z

4 a b c d e f g h i j k l m n o p q r s t u v w x y z

5 a b c d e f g h i j k l m n o p q r s t u v w x y z

Total de repeticiones 5 \* 26

Un ciclo puede contener otro ciclo, se dice que están “anidados”, estos encajes se establecen con los tres ciclos vistos. Un For puede contener otro For, un While o Repeat, todas las combinaciones están permitidas y deben responder al control de las repeticiones que requiere la solución.

Ejemplo 14.- Leer un número (N), calcular e informar la suma de los números naturales hasta ese número ( $1+2+3+...N-1+N$ ), aplicar dicho proceso sobre un conjunto cuya cantidad no se conoce. Dar por terminada la entrada cuando el número ingresado sea 0.

Al final del proceso informar cuantos números se procesaron y cuál fue el máximo.

Ejemplo: 5 6 4 0 → 15 21 10 se procesaron 3 números y el máximo es 6

Program Ej14;

Var

N, i, Max, Cont : byte;

Sum : word;

*{es importante analizar donde se inician contador, sumador y máximo}*

Begin

**Max:=0; Cont :=0;**

Write ('Ingrese un numero, 0 para finalizar'); Readln(N);

While N <> 0 do

Begin

**Sum :=0;**

For i:=1 to N do

Sum := Sum + i;

Writeln('La suma de los primeros ',N,' naturales es:', Sum);

Cont := Cont +1;

If N > Max then

Max := N;

End;

Writeln('Se procesaron ',Cont,' números y el máximo fue ',Max);

End.

Ejemplo15- En un torneo de pesca, compiten 3 categorías. Se capturan N especies diferentes registrándose por cada una:

- Código de especie (cadena de 2 caracteres) y en cada una:

✓ Categoría (1,2 ó 3 ; 0 es fin de datos)

✓ Peso

Se pide desarrollar un programa Pascal que resuelva:

- a) Para cada especie, total de ejemplares por categoría
- b) Peso promedio de toda las capturas del torneo
- c) Cantidad total de ejemplares de mas de 2 kilos

Resolver previamente en forma manual.

N = 4
TB
1 1.6
1 2.6
1 3.7
0
PR
3 2.0
1 1.9
2 1.8
2 2.0
0
CV
2 2.2
2 2.1
0
LG
1 1.5
0

Program Pesca;

Var

N, i, Cont1, Cont2, Cont3, Cont, Mas2K : Word; CodCat : byte;

CodEsp : string[2];

Peso, SumPeso : real;

Begin

Cont:=0; SumPeso:=0; Mas2K:=0;

Write(Ingrese la cantidad de distintas especies capturadas '); Readln(N);

For i:=1 to N do

Begin

Cont1:=0; Cont2:=0; Cont3:=0;

Write('Especie : '); readln(CodEsp);

Write('Categoria : '); readln(CodCat);

While CodCat <> 0 do

Begin

Write ('Peso : '); Readln(Peso);

Case CodCat of

1: Cont1:=Cont1 +1;

2: Cont2:=Cont2 +1;

3: Cont3:=Cont3 +1;

End;

SumPeso:=SumPeso + Peso;

If Peso < 2 then

Mas2K:=Mas2K + 1;

Write('Categoria : '); readln(CodCat);

End;

Cont:= Cont + Cont1 + Cont2 + Cont3;

Writeln('Especie ',CodEsp);

Writeln(' Total de Cat. 1 :',Cont1,' Total de Cat. 2 :',Cont2,' Total de Cat. 3 :',Cont3);

End;

If Cont > 0 then

Writeln('Peso promedio de ejemplares: ', SumPeso / Cont :8:2 );

Writeln('Sobre un total de ', Cont,' capturas se registraron ', Mas2K,' capturas con menos de 2 Kg de peso')

End.



Resolver los ítems 11 a 13 de los ejercicios propuestos al final del capítulo.

**Ejercitación**

1.-Leer un número N, y a continuación N números reales, calcular e informar el promedio de los mismos, descartar los ceros (si hubiera).

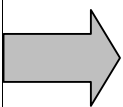
2.- Para N personas se registró el peso y la altura, calcular sus IMC y determinar e informar el % de los que tienen un índice normal ( $18 \leq \text{ICM} \leq 21$ ).

3.-Una farmacia vende medicamentos con descuento del 25% , en perfumería realiza descuento del 10% por compras mayores a \$300, en venta libre no hace descuentos.

Se pide leer los datos de K ventas, de cada uno se conoce:

- Rubro (M, P, L)
- Importe

Calcular e informar el total efectuado en concepto de descuentos y el promedio de las ventas libres. Ejemplo: K=6

M	300		Total descuentos:	165	(75 + 40 + 50 )
L	30				
P	400		Promedio venta libre:	40	
P	150				
L	50				
M	200				

4.- De M empleados se conoce la edad y la antigüedad en la empresa, a partir de dichos datos calcular e informar la antigüedad del más joven y la edad del más antiguo.

5.-Se lee un conjunto de números positivos (el último es un negativo). Se pide evaluar y mostrar para cada valor ingresado, el valor de y, siendo  $y = x + x/2 + 4$ .

6.-Se tienen pares de números reales que corresponden a largo y ancho de rectángulos, se pide ingresarlos confeccionando un listado que informe: el largo, el ancho y el área de aquellos rectángulos cuyo perímetro resulte mayor que un valor K (leído previamente), indicar además qué porcentaje de los rectángulos cumplieron lo pedido. Detener el proceso cuando ingrese el par (0, 0).

7.-Se conoce el radio de M esferas, se pide ingresar esta información para calcular e informar el volumen de cada una, detener el proceso cuando se hayan procesado los M valores o cuando el volumen de a esfera sea menor a X (ingresado por teclado).

8.- Cambiar los siguientes programas Pascal para que el ingreso de datos se realice desde archivo de texto.

- ejemplo 4e

- ejercicio 6

9.- Una consultora evalúa la aceptación de un producto en el mercado y ha grabado en un archivo de texto las respuestas de un conjunto de personas, una línea por cada una:

Edad      Respuesta (C- lo compra, I - le interesa, N-no lo conoce)

Se pide leer el archivo calcular e informar:

- a) Total de encuestados
- b) % de personas que lo compran con edad entre 20 y 45 años
- c) % de personas que le interesa sobre el total de encuestados

10.-Al finalizar un triatlón se registra de cada participante el nombre (\*\* fin de datos), el tiempo y el % de peso que perdió en la competencia.

Se pide desarrollar un programa Pascal que lea de teclado los datos mencionados, obtenga e informe:

- a) La cantidad de participantes en cada nivel
- b) El porcentaje de participantes con impacto físico leve
- c) Tiempo promedio de participantes con impacto físico moderado
- d) Cantidad de participantes de nivel 1 con impacto físico alto

Calcule el impacto físico y el nivel según las siguientes tablas:

%	Impacto físico
hasta 5	L - leve
Entre 6 y 9	M - moderado
Más de 10	A - alto

Tiempo	Nivel
Hasta 10 hs	1
Entre 11 y 20 hs	2
Mas de 20 hs	3

11.- En un hospital se hace un control diario sobre los pacientes internados, registrando su temperatura cada hora. De cada paciente se tiene el nombre ('ZZZ' fin de datos) y luego las 24 mediciones. Se pide ingresar dicha información para calcular y mostrar de cada paciente su nombre, su temperatura media y mínima. Indicar además, el nombre del paciente que registró la máxima temperatura promedio.

12.- Para cada uno de N vendedores, se conoce el Nombre, la cantidad de ventas y los respectivos importes. (ver ejemplo Capítulo 1)

Además se sabe el monto de venta fijado como objetivo para que un vendedor obtenga una recompensa. (la suma de todas sus ventas alcanza o supera dicho monto).

El orden de los datos es N, MontoObjetivo y luego por cada vendedor su nombre cantidad de ventas y los importes.

Se pide leer los datos descriptos para calcular e informar

- a) Nombre e monto Total vendido del “mejor vendedor” (mayor acumulación de importes)
- b) los Nombres de los vendedores que merecen una recompensa.
- c) Nombre del vendedor que registró la venta promedio mínima

13.- Una empresa transportista organiza sus envíos agrupándolos por código de destino y enviando un camión a cada uno. La información que se tiene es:

- Código de destino (no hay dos códigos iguales y \*\*\* indica FIN de datos)
- Capacidad de carga del camión en toneladas
- Cantidad de paquetes a transportar a este destino y el peso de cada uno en kilos.

No se sabe si todos los paquetes especificados pueden ser transportados por el camión al correspondiente destino, teniendo en cuenta que no puede sobrepasarse la capacidad del camión y que los paquetes no se pueden partir, se pide:

- a) Ingresar los datos según la descripción anterior
- b) Informar para cada uno de los destinos, cuántos kilos se cargaron en el camión
- c) Total de kilos que no pudieron ser transportados a los destinos correspondientes.
- d) Informar el código de destino cuyo camión registro la menor capacidad ociosa