

{PRACTICA1 DE LA UNIDAD1

Este programa sirve para ejemplificar lo que ocurre cuando se exceden los límites de los enteros durante una asignación (:= , readln).

Observemos que al tratar de almacenar un valor que exceda uno de los límites, el valor que se almacenar no es el correcto.

Ejemplificaremos con variables tipo BYTE (enteros en el rango 0 --> 255).

Este tipo de enteros se almacenan en 1 byte.

El cero decimal se almacena internamente como el binario 0000 0000

El 255 decimal se almacena como el binario 1111 1111.

Realicemos una operación como la siguiente (siendo B1 una variable tipo byte):

B1:=250+10

250 equivale a 1111 1010

+ 10 equivale a + 0000 0000

260 1 0000 0100

_____ /

Como B1 es una variable Estos ocho bits son de tipo BYTE solo puede los únicos que se almacenar un valor representado conservan por lo que en ocho bits por lo que éste el resultado de la bit no se almacena y se pierde operación ser 4

NOTA: Si el resultado de la operación lo almacenamos en una variable que tenga un espectro de valores suficiente

como INTEGER o WORD, por ejemplo, se conservar en forma completa.

}

program p3u3;

uses wincrt;

var

b1,b2,b3:byte;

i1:integer;

w:char;

begin

clrscr;

write('Dame un valor');

readln(b1);

write('Escribe otro valor');

readln(b2);

b3:=b1+b2;

i1:=b1+b2;

writeln('valor almacenado en la variable tipo byte: ',b3);

writeln('valor almacenado en la variable tipo entero: ',i1);

w:=readkey;

end.

{PRACTICA 2 DE LA UNIDAD 1

TIPOS STRING Y CHAR.

Estos tipos de datos permiten almacenar letras, dígitos, espacios, caracteres especiales y en general cualquier caracter de la tabla ASCII.

Los datos de estos tipos pueden asignarse mediante una instrucción **READLN** o mediante **:=** (de la misma manera que los tipos numéricos). Cuando la asignación es interna (**:=**) el dato debe encerrarse entre apóstrofes (').

Además con estos datos puede realizarse lo que se conoce como concatenación (**+**), que consiste en unir varios datos de estos tipos para formar un dato nuevo único.

Practique con el programa que tenemos a continuación.

```
}  
  
program string_y_char;  
  
uses wincrt;  
  
var  
  
c1,w:char;  
  
s1:string[20];  
s2:string[12];  
s3:string[34];  
  
pila,paterno,materno,completo1,completo2:string;  
  
begin  
  
clrscr;  
  
c1:='*';  
  
s1:=' Este mensaje estaba';  
  
s2:='en pedazos *';  
  
s3:=c1+s1+' '+s2;  
  
writeln(s3);  
  
writeln('Escribe tu nombre de pila: ');  
  
readln(pila);
```

```
writeln('Escribe tu apellido paterno: ');  
readln(paterno);  
writeln('Escribe tu apellido materno: ');  
readln(materno);  
completo1:=pila+' '+paterno+' '+materno;  
completo2:=paterno+' '+materno+' '+pila;  
writeln('Tu nombre completo es: ',completo1);  
writeln('Tu nombre por apellidos es: ',completo2);  
w:=readkey;  
end.
```

{PRACTICA 3 DE LA UNIDAD 1

MAS SOBRE DATOS STRING Y CHAR.

Si tratamos de almacenar un valor **STRING** de longitud mayor que la definida para la variable, sólo se almacenarán los primeros caracteres (según el caso). De la misma manera, si se trata de almacenar un dato de longitud mayor a un caracter en una variable de tipo **CHAR**, sólo se almacenará el primero de los caracteres.

Si un dato **STRING** contiene espacios al inicio o al final, también serán almacenados, ya que igual que cualquier otro caracter de la tabla **ASCII**, ocupan un byte de la memoria.

Se puede hacer referencia a un caracter individual de un **STRING** si se escribe el nombre de la variable y entre paréntesis cuadrados la posición correspondiente al caracter deseado. La posición cero de un **STRING** corresponde al número de caracteres que contiene el dato almacenado en la variable.

```

}

program string_y_char2;

uses wincrt;

var

c,w:char;

frase:string[10];

posicion:integer;

begin

clrscr;

write('Escribe uno o más caracteres: ');

readln(c);

writeln('El dato que quedó almacenado en la variable es: ',c);

write('Escribe una frase: ');

readln(frase);

writeln('La frase almacenada es: ',frase);

writeln('El primer caracter de la frase es: ',frase[1]);

write('¿Que posición deseas del dato? ');

readln(posicion);

write('El dato de la posición ',posicion,' es ',frase[posicion]);

w:=readkey;

end.

```

{PRACTICA 4 DE LA UNIDAD 1

Objetivo.– Adquirir los conocimientos básicos respecto a evaluación de expresiones aritméticas.

Medio.– Prueba de los operadores de división.

```

}

```

```

program operadores_de_division;

uses crt;

var

w:char;

entero1,entero2,entero3:integer;

real1,real2,real3:real;

begin

clrscr;

write('Escribe un valor entero: ');

readln(entero1);

write('Escribe otro valor entero: ');

readln(entero2);

entero3:=entero1 div entero2;

writeln('Divisi n ENTERA almacenada en variable INTEGER: ',entero3);

real2:=entero1 div entero2;

writeln('Divisi n ENTERA almacenada en variable REAL: ',real1);

real1:=entero1 / entero2;

writeln('Divisi n REAL almacenada en variable REAL: ',real1);

{La siguiente expresi n generaría un error debido a que

se emplea un operador REAL (que genera un resultado REAL)

y el resultado se trata de asignar a una variable ENTERA.

COMPRUEBELO!

entero3:=entero1 / entero2;

}

```

{Las siguientes expresiones generan error debido a que se intenta usar un operador ENTERO a una expresi n que involucra

operandos REALES:

real3:= real1 div entero1;

real3:=real1 mod entero1;

}

w:=readkey;

end.

{PRACTICA 5 DE LA UNIDAD 1

**Objetivo.– Adquirir los conocimientos básicos respecto a evaluación
de expresiones aritméticas.**

Medio.– Uso de paréntesis}

program uso_de_parenthesis;

uses wincrt;

var

r1,r2,r3,r4,r5,r6,r7,r8:real;

w:char;

x,y,z:integer;

begin

clrscr;

write('ESCRIBE EL VALOR DE X: ');

readln(x);

write('ESCRIBE EL VALOR DE Y: ');

readln(y);

write('ESCRIBE EL VALOR DE Z: ');

readln(z);

r1:=x+y*z;

r2:=x+(y*z);

```

r3:=(x+y)*z;

r4:=x+y/z;

r5:=(x+y)/z;

r6:=x/y*z;

r7:=(x/y)*z;

r8:=X/(Y*Z);

writeln('EL RESULTADO DE X+Y*Z ES: ',r1:10:4);

writeln('EL RESULTADO DE X+(Y*Z) ES: ',r2:10:4);

writeln('EL RESULTADO DE (X+Y)*Z ES: ',r3:10:4);

writeln('EL RESULTADO DE X+Y/Z ES: ',r4:10:4);

writeln('EL RESULTADO DE (X+Y)/Z ES: ',r5:10:4);

writeln('EL RESULTADO DE X/Y*Z ES: ',r6:10:4);

writeln('EL RESULTADO DE (X/Y)*Z ES: ',r7:10:4);

writeln('EL RESULTADO DE X/(Y*Z) ES: ',r8:10:4);

w:=readkey;

end.

```

{PRACTICA 6 DE LA UNIDAD 1

**Objetivo.– Aprender a utilizar algunas de las funciones básicas
interconstruidas de pascal}**

```

program funciones_basicas;

uses wincrt;

var

s1,s2:string;

r1,r2:real;

b1:byte;

c1,c2,w:char;

```

```

begin

clrscr;

{Funciones realizadas con datos string}

write('ESCRIBE UNA FRASE: ');

readln(s1);

writeln('LA LONGITUD DE LA FRASE ES: ',length(s1));

s2:=copy(s1,2,3);

writeln('TRES CARACTERES APARTIR DEL SEGUNDO SON: ',s2);

writeln;

{Funciones relacionadas con datos char}

write('ESCRIBE EL ASCII DECIMAL DE ALGUN CARACTER: ');

readln(b1);

writeln('EL CARACTER EQUIVALENTE ES: ',chr(b1));

write('ESCRIBE UN CARACTER CUALQUIERA: ');

readln(c1);

writeln('EL ASCII EQUIVALENTE ES: ',ord(c1));

write('ESCRIBE UNA LETRA MINUSCULA: ');

readln(c2);

writeln('SU EQUIVALENTE EN MAYUSCULA ES: ',upcase(c2));

writeln;

{Funciones realizadas con datos numericos}

write('ESCRIBE EL VALOR NUMERICO REAL,POSITIVO O NEGATIVO: ');

readln(r1);

writeln('SU CUADRADO ES: ',sqr(r1):15:7);

writeln('SU VALOR ABSOLUTO ES: ',abs(r1):15:7);

writeln('SU EQUIVALENTE TRUNCADO ES: ',trunc(r1):15);

```

```
writeln('SU EQUIVALENTE REDONDEADO ES: ',round(r1):15);
```

{Si r1 contiene un valor negativo, la instrucción siguiente ocasiona un error (Tome nota mental del mensaje), además, observe los resultados de las instrucciones anteriores}

```
r2:=sqrt(r1);
```

```
writeln('SU RAIZ CUADRADA ES: ',r2:15:7);
```

```
w:=readkey;
```

```
end.
```

{PRACTICA 7 DE LA UNIDAD 1

Este programa es para ejemplificar todas las funciones aritmeticas que nos brinda Pascal para el manejo de datos numericos}

```
program funciones_aritmeticas;
```

```
uses wincrt;
```

```
var
```

```
numente:integer;
```

```
numreal:real;
```

```
w:char;
```

```
const
```

```
linea='=====
```

```
begin
```

```
clrscr;
```

```
writeln(linea);
```

```
writeln('Este programa calculara todas las funciones aritméticas a partir de 2 datos ':76);
```

```
writeln(linea);
```

```
writeln('Introduce 2 valores entero/real: ':42);
```

```
readln(numente,numreal);
```

```

writeln(' I--ENTERO--I I--REAL--I ');

writeln('El valor ABSOLUTO de ambos numeros es: ',ABS(numente):15,ABS(numreal):22);

writeln('El ARCO TANGENTE de los numeros es: ',ARCTAN(numente):15,ARCTAN(numreal):22);

writeln('El COSENO de cada uno de los valores es: ',COS(numente):15,COS(numreal):22);

writeln('El EXPONENCIAL de los numeros es: ',EXP(numente):15,EXP(numreal):22);

writeln('El valor FRACCIONARIO de los datos es: ',FRAC(numente):15,FRAC(numreal):22);

writeln('El valor solo ENTERO de los datos es: ',INT(numente):15,INT(numreal):22);

writeln('El valor ALEATORIO a partir del dato es: ',RANDOM(numente):15);

writeln('El REDONDEO de los numeros es: ',ROUND(numente):15,ROUND(numreal):22);

writeln('El SENO calculado por cada numero es: ',SIN(numente):15,SIN(numreal):22);

writeln('El CUADRADO de ambos numeros dados es: ',SQR(numente):15,SQR(numreal):22);

writeln('La RAIZ CUADRADA de los numeros es: ',SQRT(numente):15,SQRT(numreal):22);

writeln('El INCREMENTO de la unidad es: ',SUCC(numente):15);

writeln('El TRUNCAMIENTO de los numeros es: ',TRUNC(numente):15,TRUNC(numreal):22);

writeln('El DECREMENTO de la unidad es: ',PRED(numente):15);

writeln('El LOGARITMO NATURAL calculado es: ',LN(numente):15,LN(numreal):22);

writeln;

writeln('OPRIME LA TECLA <ENTER> PARA CONTINUAR...':79);

w:=readkey;

end.

```

{PRACTICA 1 DE LA UNIDAD 2

Hasta ahora, los ejercicios que hemos realizado son programas

muy sencillos: el orden de ejecución de las instrucciones es

el básico, de la primera a la ultima según su orden físico.

Pero a partir de ahora debemos aprender a programar soluciones

a problemas que requieren el uso de instrucciones que alteran

el orden natural de ejecución de un programa.

La primera instrucción que aprenderemos es la instrucción IF.

Esta instrucción equivale al símbolo:

Escribamos un programa que resuelva lo siguiente:

Supongamos que en nuestro país las empresas deben pagar sus impuestos dependiendo del monto de sus utilidades. Si la empresa tiene utilidades mayores a \$50,000.00 el Impuesto a pagar será un 40 % de las utilidades, de lo contrario, el Impuesto será un 35 % de las utilidades de la empresa y dar como resultado el Impuesto que debe pagar.

Nota: Este ejemplo muestra el "IF LOGICO" (por cada salida un proceso).

Maestro: Dibuje el Diagrama de flujo.

La sintaxis de IF es la siguiente:

IF expresión booleana THEN

instrucción

ELSE

instrucción;

Una expresión booleana es aquella que da como resultado FALSO o VERDADERO.

Los operadores básicos que se usan en estas expresiones son:

< , > , = , <= , >= , <> }

program if_then_else;

uses wincrt;

var

utilidad,impuesto,porcimp:real;

```

begin
clrscr;

writeln('** Este programa calcula el Impuesto **');

writeln(' que deben pagar las empresas. ');

writeln;

write('Escriba la utilidad de la empresa: ');

readln(utilidad);

writeln;

if utilidad>50000.00 then

porcimp:=0.4

else

porcimp:=0.35;

impuesto:=utilidad*porcimp;

writeln('El impuesto a pagar es: ',impuesto:10:2);

readkey;

end.

```

{PRACTICA 2 DE LA UNIDAD 2

Más ejercicios con IF_THEN_ELSE

Este programa calcula el sueldo semanal de un empleado

Cada hora tiene un valor de 4 pesos, si pasa de 40 horas

Laboradas, la hora extra tiene un valor de 6 pesos}

```
program empleado;
```

```
uses wincrt;
```

```
var
```

```
ht,sueldot:real;
```

```
w:char;
```

```

begin

clrscr;

writeln('** este programa calcula el sueldo semanal de un empleado ** ');

writeln;

write('escriba las horas trabajadas en la semana');

readln(ht);

if ht<=40 then

sueldot:=ht*4.00

else

sueldot:=(40*4.00)+((ht-40)*6.00);

writeln('el sueldo a pagar es: ', sueldot:7:2);

w:=readkey;

end.

```

{PRACTICA 3 DE LA UNIDAD 2

**Este programa calcula cuantos billetes de 1000, 500, 200, 100, 50
y cuantas monedas de 10, 5 y 1 peso existen en una cantidad}**

```

program cantidad;

uses wincrt;

var

cant1:longint;

cant2,cant3,cant4,cant5,cant6,cant7,cant8,cant9,cant10:integer;

w:char;

begin

clrscr;

write('Dame la cantidad a calcular: ');

readln(cant1);

```

```
if cant1>=1000 then
begin
cant2:=cant1 div 1000;
cant1:=cant1 mod 1000;
end;

if cant1>=500 then
begin
cant3:=cant1 div 500;
cant1:=cant1 mod 500;
end;

if cant1>=200 then
begin
cant4:=cant1 div 200;
cant1:=cant1 mod 200;
end;

if cant1>=100 then
begin
cant5:=cant1 div 100;
cant1:=cant1 mod 100;
end;

if cant1>=50 then
begin
cant6:=cant1 div 50;
cant1:=cant1 mod 50;
end;

if cant1>=20 then
```

```

begin

cant7:=cant1 div 20;

cant1:=cant1 mod 20;

end;

if cant1>=10 then

begin

cant8:=cant1 div 10;

cant1:=cant1 mod 10;

end;

if cant1>=5 then

begin

cant9:=cant1 div 5;

cant1:=cant1 mod 5;

end;

if cant1>=1 then

begin

cant10:=cant1 div 1;

cant1:=cant1 mod 1;

end;

writeln('La cantidad en billetes de a 1000 es: ', cant2);

writeln('La cantidad en billetes de a 500 es: ', cant3);

writeln('La cantidad en billetes de a 200 es: ', cant4);

writeln('La cantidad en billetes de a 100 es: ', cant5);

writeln('La cantidad en billetes de a 50 es: ', cant6);

writeln('La cantidad en monedas de a 20 es: ', cant7);

writeln('La cantidad en monedas de a 10 es: ', cant8);

```

```
writeln('La cantidad en monedas de a 5 es: ', cant9);

writeln('La cantidad en monedas de a 1 es: ', cant10);

w:=readkey;

end.
```

{PRACTICA 4 DE LA UNIDAD 2

Este programa calcula el costo del porte de una carta

Si la peso es menor o igual a 100 gramos se multiplicará por 0.29

En caso contrario cada gramo extra se multiplicara por 0.04}

```
program costo_carta;

uses wincrt;

var

peso,costo:real;

w:char;

begin

clrscr;

writeln('Cual es el peso de la carta (la medida es en gramos): ');

readln(peso);

if peso<=1then

costo:=peso*0.29

else

costo:=(1*0.29)+((peso-1)*0.04);

writeln('El costo del peso de la carta es: ', costo:8:2);

w:=readkey;

end.
```

{PRACTICA 5 DE LA UNIDAD 2

Este programa calcula el costo de la prima de un seguro

Si es varón y la edad es menor a 25 pagará 1000 pesos

Si es varón y la edad es mayor o igual a 25 pagará 700 pesos

Si es mujer y la edad menor a 21 pagará 800 pesos

Si es mujer y la edad es mayor o igual a 21 pagará 500 pesos}

program seguros;

uses wincrt;

var

s,w:char;

e,p:integer;

begin

clrscr;

writeln('Cual es tu sexo (V)aron o (M)ujer: ');

readln(s);

writeln('Cual es tu edad ');

readln(e);

if s='v' then

if e<25 then

p:=1000

else

p:=700

else

if e<21 then

p:=800

else

p:=500;

writeln('La prima a pagar es: ', p);

w:=readkey;

end.

{PRACTICA 6 DE LA UNIDAD 2

Este program escribe el código fuente para imprimir la letra correspondiente a las calificaciones de los estudiantes}

program cal;

uses wincrt;

var

l,w:char;

cal1:real;

begin

clrscr;

writeln('Cual es la calificación obtenida (0-100): ');

readln(cal1);

if cal1 >= 90 then

l:='a'

else

if (cal1>=80) and (cal1<90) then

l:='b'

else

if (cal1>=70) and (cal1<80) then

l:='c'

else

if (cal1>=60) and (cal1<70) then

l:='d'

else

if cal1<60 then

```

l:='e';

writeln('La calificación del alumno es: ', l);

w:=readkey;

end.

```

{PRACTICA 7 DE LA UNIDAD 2

Objetivo.- Conocimiento de la instrucción CASE expresión OF

Sintáxis.- CASE expresion OF

Caso 1: instrucción;

Caso 2: instrucción;

.....

ELSE instrucción;

END;

Notas.- Los casos pueden ser una lista de valores o expresiones

Separadas por comas o un rango separado por puntos suspensivos}

```

program case_of;

uses wincrt;

var

i:integer;

c:char;

begin

clrscr;

write('Escribe un valor numérico: ');

readln(i);

case i of

-32768..-1: writeln('Escribiste un valor negativo: ');

1..32767: writeln('Escribiste un valor positivo: ');

```

```

0 : writeln('Escribiste un cero ');

{en este caso podemos usar ELSE}

end;

writeln;

writeln('Presiona cualquier tecla: ');

c:=readkey;

case c of

chr(7), chr(13) : begin

writeln('No estoy seguro ');

writeln('si presionaste ');

writeln('ESCAPE o ENTER ');

end;

chr(8) :writeln('Presionaste BACKSPACE ');

'A'..'Z' :writeln('Presionaste una LETRA MAYUSCULA ');

chr(97).. chr(122) :writeln('Presionaste una letra minúscula ');

'0'..'9' :writeln('Presionaste un DIGITO ');

else

writeln('NO SE QUE PRESIONASTE ');

end;

end.

```

{PRACTICA 8 DE LA UNIDAD 2

Este programa calcula los datos de los lados desconocidos de un triángulo rectángulo según el teorema de pitágoras}

```

program pitagoras;

uses wincrt;

var

a,b,c,ca,cb,cc:real;

```

```

od:byte;

w:char;

begin

clrscr;

writeln('**Triangulo Rectángulo** ');

writeln('1.-Primer caso (calcular hipotenusa) ');

writeln('2.-Segundo caso (calcular cateto) ');

writeln('3.-ninguno ');

write('Seleccione la opción deseada (1,2,3): ');

readln(od);

case od of

1:begin

writeln('Escribe la medida del cateto menor: ');

readln(a);

writeln('Escribe la medida del cateto mayor: ');

readln(b);

ca:=sqr(a);

cb:=sqr(b);

c:=sqrt(ca+cb);

writeln('La hipotenusa tiene una medida de: ', c:15:2);

end;

2:begin

writeln('Escribe la medida de la hipotenusa: ');

readln(c);

writeln('Escribe la medida del cateto: ');

readln(a);

```

```

cc:=sqr(c);

ca:=sqr(a);

b:=sqrt(c-a);

writeln('La medida del otro cateto es: ', b:15:2);

end;

3:begin

writeln('**fin del programa** ');

end;

end;

w:=readkey

end.

```

{PRACTICA 9 DE LA UNIDAD 2

Desde luego podemos implementar contadores con los ciclos REPEAT y WHILE, pero la forma más eficiente y sencilla en Pascal es mediante un ciclo FOR.

La sintaxis es la siguiente:

```

FOR <variable ordinal>:=<inicial> TO ó DOWNTO <final> DO
instrucción;

```

Los valores ordinales son aquellos que solo tienen un valor sucesor y uno predecesor, por ejemplo, los valores INTEGER, CHAR, SHORTINT, LONGINT, BYTE, WORD, incluso BOOLEAN, etc.

```

}

```

```

program ciclo_for;

uses wincrt;

var

i:integer;

```

```

c:char;

begin

clrscr;

writeln('A continuación se imprimen los números del 1 al 10: ');

for i:=1 to 10 do

writeln(i);

writeln('Presione cualquier tecla . . .');

readkey;

writeln('Ahora al revés: ');

for i:=10 downto 1 do

writeln(i);

writeln('Presione cualquier tecla . . .');

readkey;

writeln('A continuación se imprimen las letras de la A a la M: ');

for c:='A' to 'M' do

writeln(c);

writeln('Presione cualquier tecla . . .');

readkey;

writeln('Ahora al revés: ');

for c:='M' downto 'A' do

writeln(c);

writeln('Presione cualquier tecla . . .');

readkey;

writeln(' A continuación imprimimos una tabla ASCII: ');

for i:=1 to 255 do

begin

```

```

if i mod 20=0 then

begin

writeln('Presione cualquier tecla . . . ');

readkey;

end;

writeln(i, ' ', chr(i));

end;

readkey;

end.

```

{PRACTICA 10 DELA UNIDAD 2

Este programa imprime la tabla del 5 por medio del ciclo for}

```

program ciclo_for;

uses wincrt;

var

x,i:integer;

w:char;

begin

clrscr;

for i:=1 to 10 do

begin

x:=5*i;

writeln('5','*',i,'=',x );

end;

w:=readkey;

end.

```

{PRACTICA 11 DELA UNIDAD 2

Este programa imprime la tabla del 5 por medio del ciclo while}

program ciclo_while;

uses wincrt;

var

i,x:integer;

w:char;

begin

clrscr;

i:=0;

while i<10 do

begin

i:=i+1;

x:=5*i;

writeln('5','*',i,'=',x);

end;

w:=readkey;

end.

{PRACTICA 12 DELA UNIDAD 2

Este programa imprime la tabla del 5 por medio del ciclo repeat}

Program ciclo_repeat;

uses wincrt;

var

x,i:integer;

w:char;

begin

clrscr;

```

i:=0;

repeat

i:=i+1;

x:=5*i;

writeln('5','*',i,'=',x );

until i=10;

w:=readkey;

end.

```

{PRACTICA13 DELA UNIDAD 2

Este programa calcula una secuencia de números

1,2,4,8,16,32,64,128 y 256 e imprimirá la suma de los números}

Program suma;

uses wincrt;

var

i,x,y:integer;

w:char;

begin

clrscr;

i:=1;

for x:=1 to 9 do

begin

y:=y+i;

write(i,',');

i:=i*2

end;

writeln('la suma es: ',y);

w:=readkey;

end.

{PRACTICA 14 DE LA UNIDAD 2

Este programa ejemplifica el uso de los ciclos REPEAT.

Estos ciclos son aquellos que en nuestros diagramas de flujo

Tienen la condición al final del cuerpo del ciclo.

Maestro dibuje un Diagrama de Flujo.

La sintaxis de la instrucción es la siguiente:

REPEAT

< instrucción>;

.....

.....

.....

< instrucción>;

UNTIL <condición>;

Unos de los ejemplos más usuales se ejemplifica a continuación}

Program validation;

uses wincrt;

var

cal:integer;

sexo,edociv:char;

begin

{Ejercicio 1: Condicionar un dato a un rango de valores

(validación)}

repeat

clrscr;

```

write('Escribe tu calificación de la unidad I: ');

readln(cal);

until (cal>=0) and (cal<=100);

{Ejercicio 2: Pausas durante un programa}

write('Presione cualquier tecla...');

repeat until keypressed;

{Ejercicio 3: Una forma más presentable de Validación}

repeat

clrscr;

gotoxy(1,10);

write('Escribe tu calificación de la unidad I: ');

readln(cal);

until (cal>=0) and (cal<=100);

{Ejercicio 4: Validación de valores CHAR o STRING}

repeat

clrscr;

gotoxy(1,12);

write('Sexo Femenino o Masculino (F/M): ');

readln(sexo);

until (sexo='F') or (sexo='M');

{Ejercicio 5: Validación de valores CHAR o STRING (una mejor)}

repeat

clrscr;

gotoxy(1,14);

write('Estado civil, Soltero o Casado (S/C): ');

readln(edociv);

```

```
edociv:=upcase(edociv);
```

```
until (edociv='S') or (edociv='C');
```

```
end.
```

{PARACTICA 15 DE LA UNIDAD 2

Este programa calcula el factorial de un numero cualquiera}

```
program factorial;
```

```
uses wincrt;
```

```
var
```

```
i,sum,num:integer;
```

```
w,resp:char;
```

```
begin
```

```
repeat
```

```
write('Número al que desea obtener el factorial: ');
```

```
readln(num);
```

```
sum:=1;
```

```
if num =0 then
```

```
sum:=1
```

```
else
```

```
begin
```

```
for i:= num downto 1 do
```

```
sum:=sum*i
```

```
end;
```

```
writeln('El Factorial de ',num, ' es ', sum);
```

```
write('Desea otro número S/N ');
```

```
readln(resp);
```

```
resp:=upcase(resp);
```

until resp='N';

w:=readkey;

end.

{PRACTICA 16 DELA UNIDAD 2

Este programa captura una serie de edades, terminará de capturar hasta que presione

La tecla 0 (cero) sin tomarla en cuenta como edad.

Imprime las edades capturadas , así como la edad mayor, menor, el promedio de edades

Y la suma de las edades.}

Program prom;

uses wincrt;

var

me,m,c,s,n:byte;

p:real;

w:char;

begin

me:=200;

m:=0;

c:=0;

s:=0;

clrscr;

repeat

gotoxy(10,4);

write('Edad: ');

gotoxy(40,4);

write('Edad 0 para salir ');

gotoxy(20,4);

```

readln(n);

if n>m then

m:=n;

if n<me then

if n>0 then

me:=n;

s:=s+n;

gotoxy(50,c+5);

write(n);

c:=c+1; gotoxy(20,4);

write(' ');

until n=0;

gotoxy(50,c+5);

write(s);

p:=s/(c-1);

gotoxy(30,12);

write('El mayor es: ', m);

gotoxy(30,13);

write('El menor es: ', me);

gotoxy(30,14);

write('El promedio es: ', p:8:2);

writeln;

w:=readkey;

end.

```

{PRACTICA 17 DE LA UNIDAD 2

Este programa captura una cantidad de N edades y maneja un rango :

1-10

11-20

21-30

31-40

>40

El programa imprime, la cantidad de edades que estuvieron en cada rango}

program edades;

uses wincrt;

var

a,b,c,d,e,f,i,n:integer;

w:char;

begin

a:=0; b:=0; c:=0; d:=0; f:=0;

clrscr;

writeln('¿Cuántas edades vas a capturar: ');

readln(n);

for i:=1 to n do

begin

writeln('Edad ',i, ' ');

readln(e);

if (e>=1) and (e<=10) then a:=a+1;

if (e>=11) and (e<=20) then b:=b+1;

if (e>=21) and (e<=30) then c:=c+1;

if (e>=31) and (e<=40) then d:=d+1;

if (e>=41) then f:=f+1;

end;

```
writeln('De 1-10: ',a);

writeln('De 11-20: ',b);

writeln('De 21-30: ',c);

writeln('De 31-40: ',d);

writeln('Mayores que 40: ',f);

w:=readkey;

end.
```

{PRACTICA 1 DE LA UNIDAD 3

ARREGLOS.

Un arreglo es una Estructura de Datos, es decir, una disposición ordenada de datos, permite almacenar elementos del MISMO TIPO y manejar el grupo de datos como un todo, o cada elemento en forma individual según se requiera.

Todas las variables que hemos estudiado hasta ahora, han sido de un tipo simple, o sea que consiste de una celda INDIVIDUAL de memoria que solo puede contener un valor a la vez, las variables del tipo ARREGLO, como lo indicamos en el párrafo anterior, pueden contener más de un valor a la vez.

Podemos enumerar las características que deben tener los arreglos y los programas que requieran uso de arreglos:

- Los arreglos constan de una colección de celdas de memoria cada una de las cuales puede contener un dato individual.
- Un arreglo puede compararse con un edificio de oficinas, el nombre del edificio equivale al NOMBRE del arreglo. Cada oficina equivale a un ELEMENTO del arreglo. Además cada oficina tiene un número asignado que corresponde a su posición dentro del edificio, y sirve como identificación de la oficina. El número de la oficina equivale al INDICE de cada elemento del arreglo.
- Durante la declaración de variables, se le debe informar a PASCAL el nombre, el tipo de los datos que va a contener y los índices inferior y superior del arreglo.
- El INDICE debe ser un valor de tipo ordinal (INTEGER, SHORTINT, CHAR, ETC.).
- Cuando se desea hacer referencia a un arreglo en forma conjunta solo se requiere indicar su NOMBRE. Si se desea en cambio, referirse a un elemento en particular, debe hacerse con el NOMBRE del arreglo seguido de paréntesis rectangulares (corchetes) y entre ellos el INDICE del ELEMENTO deseado.
- Se recomienda usar la Directiva del Compilador {\$R+} que permite que PASCAL responda con un mensaje de error cuando se intente hacer referencia a un elemento del arreglo, con un índice fuera del rango declarado.

Si no usamos esa directiva, Pascal nunca revisa que el índice sea válido antes de calcular una dirección de memoria equivalente al índice, por lo que si éste no está en el rango definido, podemos destruir (si hacemos una asignación) un valor almacenado en la memoria correspondiente a otra variable o al programa mismo. El daño mínimo puede consistir en considerar (al consultar, usar, etc) como perteneciente al arreglo un valor que no lo es. En este programa al final realizamos un pequeño ejercicio para comprobar lo anterior.

- Los arreglos pueden tener más de una expresión de índice, por ejemplo, cuando se desea manejar en un arreglo una matriz, se puede definir una variable tipo ARRAY con dos expresiones de índice.}

```
program arreglos;

uses wincrt;

var

nombre:array[1..30] of char;

c,i:integer;

resp:char;

begin

clrscr;

write(' Escribe tu nombre (coloca un punto al final para finalizar): ');

c:=0;

repeat

c:=c+1;

nombre[c]:=readkey;

write(nombre[c]);

until nombre[c]='.';

writeln;

writeln('La primera letra de tu nombre es: ',nombre[1]);

write('Deseas consultar una posición de tu nombre S/N?: ');

resp:=upcase(readkey);

while resp='S' do

begin

writeln('Qué posición deseas? ');readln(i);

writeln('El elemento de la posición ',i,' es ',nombre[i]);

writeln('Deseas consultar otra posición de tu nombre S/N? ');

resp:=upcase(readkey);
```

end;

end.

{PRACTICA 2 DE LA UNIDAD 3

El siguiente programa captura una lista de N calificaciones, calcula el promedio de ese

Grupo, imprime cuantas calificaciones hubo mayores que el promedio}

{ \$R+ }

program prom2;

uses wincrt;

var

cal:array[1..100]of integer;

i,c,sum,n:integer;

w:char; prom:real;

begin

clrscr;

write('De cuantas personas es el grupo: ');

readln(n);

sum:=0;

for i:=1 to n do

begin

write('Dame la calificación: ');

readln(cal[i]);

sum:=sum+cal[i];

end;

prom:=sum/n;

c:=0;

for i:=1 to n do

```

begin

if cal[i]>prom then

c:=c+1;

end;

writeln('El promedio es: ',prom:4:2);

writeln('Se encontraron ',c, ' calificaciones mayores que el promedio ');

w:=readkey;

end.

```

{PRACTICA 3 DE LA UNIDAD 3

El siguiente programa captura una frase cualquiera que se desee

Imprimirá la cantidad de letras que existien en esa frase}

{ \$R+ }

```

program letras;

uses wincrt;

var

frase:string;

i:integer;

c,w,una letra:char;

cont:array ['A'..'Z'] of byte;

begin

clrscr;

for c:='A' to 'Z' do

cont [c]:=0;

write('Escribe una frase: ');

readln(frase);

for i:=1 to length(frase)do

```

```

begin
unaletra:=upcase(frase[i]);
if(unaletra>='A') and (unaletra<='Z') then
inc(cont[unaletra]);
end;
for c:='A' to 'Z' do
begin
if cont[c]<>0 then
writeln(cont[c], ' Letras ',c);
end;
w:=readkey;
end.

```

{PRACTICA 4 DE LA UNIDAD 3

Este programa realiza la suma de 2 vectores de igual dimensión e imprime el resultado}

{ \$R+ }

```

program suma;
uses wincrt;
var
i:integer;
a,b,c:array [1..5] of integer;
w:char;
begin
clrscr;
for i:=1 to 5 do
begin
write('Dame el valor ',i, 'del vector a ');

```

```

readln(a[i]);

end;

clrscr;

for i:=1 to 5 do

begin

write('Dame el valor ' ,i, ' del vector b ');

readln(b[i]);

end;

for i:=1 to 5 do

c[i]:=a[i]+b[i];

gotoxy(19,8);write('El resultado de la suma de los vectores es: ');

writeln;writeln;writeln;

for i:=1 to 5 do

begin

gotoxy(20+3*i,12);write(c[i]);

end;

w:=readkey;

end.

```

{PRACTICA 5 DE LA UNIDAD 3

Este programa realiza la resta de 2 vectores de igual dimensión e imprime el resultado}

{ \$R+ }

```

program resta;

uses wincrt;

var

i:integer;

a,b,c:array [1..5] of integer;

```

```

w:char;

begin

clrscr;

for i:=1 to 5 do

begin

write('Dame el valor ' ,i, 'del vector a ');

readln(a[i]);

end;

clrscr;

for i:=1 to 5 do

begin

write('Dame el valor ' ,i, ' del vector b ');

readln(b[i]);

end;

for i:=1 to 5 do

c[i]:=a[i]-b[i];

gotoxy(19,8);write('El resultado de la resta de los vectores es: ');

writeln;writeln;writeln;

for i:=1 to 5 do

begin

gotoxy(20+3*i,12);write(c[i]);

end;

w:=readkey;

end.

```

{PRACTICA 6 DE LA UNIDAD 3

Este programa realiza la multiplicación de 2 vectores de igual dimensión e imprime el resultado}

```

{$R+}

program multiplicacion;

uses wincrt;

var

i,p:integer;

a,b,c:array [1..5] of integer;

w:char;

begin

clrscr;

for i:=1 to 5 do

begin

write('Dame el valor ' ,i, 'del i vector ');

readln(a[i]);

end;

clrscr;

for i:=1 to 5 do

begin

write('Dame el valor ' ,i, ' del i vector ');

readln(b[i]);

end;

clrscr;

gotoxy(22,4);write('El producto de los vectores es: ');

writeln;writeln;

p:=0;

for i:=1 to 5 do

p:=p+a[i]*b[i];

```

```
gotoxy(36,8);write(p);
```

```
w:=readkey;
```

```
end.
```

```
{PRACTICA 7 DE LA UNIDAD 3
```

```
Este programa genera una matriz de 5 x 5 }
```

```
{ $R+ }
```

```
program matrices;
```

```
uses wincrt;
```

```
var
```

```
i,j:integer; w:char;
```

```
num:array[1..5,1..5] of integer;
```

```
begin
```

```
clrscr;
```

```
for i:=1 to 5 do
```

```
for j:=1 to 5 do
```

```
begin
```

```
write('Número: ',i,' ',j,' ');
```

```
readln(num[i,j]);
```

```
end;
```

```
clrscr;
```

```
for i:=1 to 5 do
```

```
for j:=1 to 5 do
```

```
begin
```

```
gotoxy(i,j+2);
```

```
write(num[i,j]);
```

```
end;
```

```
w:=readkey;
```

```
end.
```

{PRACTICA 8 DE LA UNIDAD 3

Este programa captura los nombres en un vector, y las calificaciones en una matriz

Imprime el nombre, sus calificaciones y el promedio por alumno}

```
{ $R+ }
```

```
program calificaciones;
```

```
uses wincrt;
```

```
var i,j:integer; sum:real; w:char;
```

```
nom:array[1..2] of string;
```

```
prom:array [1..2] of real;
```

```
cal:array[1..2,1..2] of real;
```

```
nombre:string;
```

```
begin
```

```
for i:=1 to 2 do
```

```
begin
```

```
clrscr;
```

```
write('Nombre: ',i, ' ');
```

```
readln(nom[i]);
```

```
sum:=0;
```

```
for j:=1 to 2 do
```

```
begin
```

```
write('Calificación: ',i, ' ',j, ' ');
```

```
readln(cal[i,j]);
```

```
sum:=sum+cal[i,j];
```

```
end;
```

```

prom[i]:=sum/2;

end;

clrscr;

for i:=1 to 2 do

begin

write(nom[i]:2);

for j:=1 to 2 do

write(cal[i,j]:5:2, ' ');

write(prom[i]:5:2);

writeln;

end;

w:=readkey;

end.

```

{PRACTICA 9 DE LA UNIDAD 3

El siguiente programa captura una lista de nombres y los ordena en forma ascendente

Por el método de burbuja, e imprime el vector ordenado}

{ \$R+ }

```

program ordenacion;

uses wincrt;

var

nom:array[1..100] of string [40];

t:string[40];

i,j,n:integer;

begin

clrscr;

write('Cuántos nombres deseas capturar ');

```

```

readln(n);

for i:=1 to n do

begin

write('Dame el nombre ',i, ' ');

readln(nom[i]);

end;

for i:=1 to n-1 do

begin

for j:=i+1 to n do

begin

if nom[i] > nom[j] then

begin

t:=nom[i];

nom[i]:=nom[j];

nom[j]:=t;

end;

end;

end;

for i:=1 to n do

writeln(nom[i]);

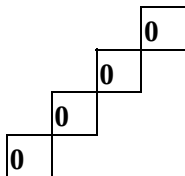
readkey;

end.

```

{PRACTICA 10 DE LA UNIDAD 3

Este programa imprime la siguiente matriz:



```

}

program diagonal_invertida;

uses wincrt;

var

i,j:integer;

w:char;

begin

clrscr;

for i:=1 to 4 do

for j:=1 to 4 do

begin

if (i+j=5) then

begin

gotoxy(i,j);

write('0');

end;

end;

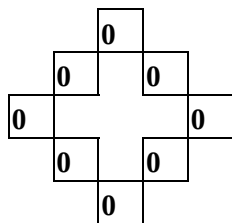
w:=readkey;

end.

```

{PRACTICA 11 DE LA UNIDAD 3

Este programa imprime la siguiente matriz:



```

}

program rombo_de_ceros;

```

```
uses wincrt;

var

i,j:integer; w:char;

begin

  clrscr;

  for i:=1 to 5 do

    for j:=1 to 5 do

      begin

        if (i+j=4) then

          begin

            gotoxy(i,j);

            write('0');

            end;

          if (i+j=8) then

            begin

              gotoxy(i,j);

              write('0');

              end;

            if (i*j=8) then

              begin

                gotoxy(i,j);

                write('0');

                end;

              end;

            end;

          w:=readkey;

        end.
```

{PRACTICA 1 DE LA UNIDAD 4

Esta función permite elevar un valor real a una potencia real.

Por ejemplo:

6.3 elevado a la 3.1

Para resolverlo emplearemos la fórmula

$\text{LogE}(3.1)*6.3$

}

program funcionpotencia;

uses wincrt;

var

w:char;

r1,r2:real;

function pot(x,y:real):real;

begin

pot:=exp(ln(x)*y);

end;

begin

clrscr;

writeln('2 Elevado al cubo es: ',pot(2,3):4:2);

r1:=pot(3,9);

writeln('3 Elevado a la potencia 9 es: ',r1:8:2);

r2:=pot(9,0.5);

writeln('La raiz cuadrada de 9 es: ',r2:4:2);

writeln('2 Elevado a la potencia 0 es: ',pot(2,0):4:2);

w:=readkey;

end.

{PRACTICA 2 DE LA UNIDAD 4

Este programa eleva al cubo cualquier número entero utilizando funciones}

```
program elevcub;  
  
uses wincrt;  
  
var  
  
x:integer; w:char;  
  
function cubo(x:integer):integer;  
  
begin  
  
cubo:=x*x*x;  
  
end;  
  
begin  
  
clrscr;  
  
writeln('Cual es el número a elevar ');  
  
readln(x);  
  
writeln('El cubo del número es: ',cubo(x));  
  
w:=readkey;  
  
end.
```

{PRACTICA 3 DE LA UNIDAD 4

Este programa convierte un valor string cualquiera en su equivalente en mayúscula.

NOTA: La solución propuesta para este problema no es satisfactoria, ya que, no convierte a mayúsculas las áéíóúñ. }

```
program funcionconvertirmayusculas;  
  
uses wincrt;  
  
var  
  
nombre:string;  
  
function funcnom(nom:string):string;  
  
var
```

```

i:integer;

begin

for i:=1 to length(nom) do

begin

nom[i]:=upcase(nom[i]);

funcnom:=nom;

end;

end;

begin

clrscr;

writeln(funcnom('Lenguaje Pascal '));

write('Escribe tu nombre: ');

readln(nombre);

writeln('Tu nombre en mayúsculas es: ',funcnom(nombre));

writeln(funcnom('Verónica Nuñez '));

readkey;

end.

```

{PRACTICA 4 DE LA UNIDAD 4

Este ejercicio ilustra la forma en que Pascal implementa la Recursividad Simple.

Un ejemplo típico de Recursividad Simple es el del cálculo del factorial de un número.

La definición del factorial de un número cualquiera (entero positivo) dice: El factorial de un número cualquiera es igual al número mismo multiplicado el factorial de su número predecesor. Además por definición el factorial de 0 es 1.

$$X! = x * (x-1)!$$

Lo anterior es una definición RECURSIVA, los lenguajes de Programación nos permiten implementar soluciones usando este tipo de definiciones. Lo único que se requiere adicionalmente a la definición recursiva, es una condición de terminación (UN VALOR CONOCIDO), en este caso, el factorial de 0 es 1.

Como lo veremos en nuestro ejemplo, dentro de la definición del subprograma, como este se LLAMA A

EJECUCION A SI MISMO; la variable asociada al nombre de una función NO PUEDE USARSE COMO ACUMULADOR, ya que Pascal entendería que se está haciendo referencia al nombre de la función en forma recursiva.

Otra forma de Recursividad, denominada por algunos autores como `COMPLEJA', es permitida por Pascal. La forma más sencilla de ésta forma de Recursividad es la siguiente (suponiendo que A y B, son subprogramas):

Sub_Prog_A llama a Sub_Prog_B y Sub_Prog_B llama a Sub_Prog_A

Recordemos que en la Recursividad Simple:

Sub_Prog_A llama a Sub_Prog_A}

program recursividad;

uses wincrt;

var

x:shortint;

function fact(x:shortint):real;

begin

if x=0 then

fact:=1

else

fact:= x * fact(x-1);

end;

var

n:shortint;

begin

clrscr;

writeln('El factorial de 0 es: ',fact(0):10:2);

writeln('El factorial de 5 es: ',fact(5):10:2);

writeln('El factorial de 7 es: ',fact(7):10:2);

write('De qué numero deseas obtener el factorial? ');

```

readln(n);

writeln('El factorial de ',n, ' es: ',fact(n):10:2);

readkey;

end.

```

{PRACTICA 5 DE LA UNIDAD 4

Este programa centra una frase, se utiliza la funcion gotoxy}

```

program cantrar;

uses wincrt;

var

frase:string; w:char;

function centrarfra(frase:string):integer;

begin

centrarfra:=(80-(length(frase)))div 2;

end;

begin

clrscr;

writeln('Escribe una frase ');

readln(frase);

gotoxy(centrarfra(frase),10);

writeln(frase);

w:=readkey;

end.

```

{PRACTICA 6 DE LA UNIDAD 4

El siguiente programa calcula el promedio y nos envía como resultado si está bien, regular o mal}

```

program enteros;

uses wincrt;

```

```

var

x,y,z:integer; w:char;

function promcal(x,y,z:integer):string;

var

prom:real; cad:string;

begin

prom:=(x+y+z)/3;

if prom>85 then cad:='bien';

if (prom>=70) and (prom<85) then cad:='regular';

if prom<70 then cad:='mal';

promcal:=cad;

end;

begin

clrscr;

writeln('Dame la calificación 1: ');

readln(x);

writeln('Dame la calificación 2: ');

readln(y);

writeln('Dame la calificación 3: ');

readln(z);

writeln('tu promedio estuvo: ',promcal(x,y,z));

w:=readkey;

end.

```

{PRACTICA 7 DE LA UNIDAD 4

MAS PRACTICAS SOBRE FUNCIONES:

Debera indicar cual es el resultado del siguiente programa}

```

program ejercicio;

uses wincrt;

var

a,b,c:integer;

function ejecuta(x,y:integer):integer;

var

w,i:integer;

begin

w:=1;

for i:=1 to y do

w:=w*x;

ejecuta:=w;

end;

begin

clrscr;

a:=3;

b:=4;

c:=ejecuta(a,b);

writeln(a,' ',b,' ',c);

b:=ejecuta(b,a);

writeln(a,' ',b,' ',c);

a:=ejecuta(a,a);

writeln(a,' ',b,' ',c);

end.

```

{PRACTICA 8 DE LA UNIDAD 4

Más prácticas sobre funciones:

Este programa realiza : la captura del día del mes de nacimiento así como el mes,

Y nos devuelve como resultado el ciclo de estación del año}

```
program estaciones;

uses wincrt;

var

m,d:integer;

w:char;

function esta(d,m:integer):string;

begin

case m of

1,2:esta:='INVIERNO';

3:begin

if d<21 then

esta:='INVIERNO'

else

esta:='PRIMAVERA';

end;

4,5:esta:='PRIMAVERA';

6:begin

if d<21 then

esta:='PRIMAVERa'

else

esta:='VERANO';

end;

7,8:esta:='VERANO';

9:begin
```

```

if d<21 then

esta:='VERANO'

else

esta:='OTOÑO';

end;

10,11:esta:='OTOÑO';

12:begin

if d<21 then

esta:='OTOÑO'

else

esta:='INVIERNO';

end;

end;

end;

begin

clrscr;

writeln('dia de nacimiento ');

readln(d);

writeln('mes de nacimiento de tipo numerico ');

readln(m);

writeln('tu naciste en: ',esta(d,m));

w:=readkey;

end.

{PRACTICA 9 DE LA UNIDAD 4

CUADERNO DE PRACTICAS

MATERIA: LENGUAJE DE PROGRAMACION I

```

UNIDAD

I

ELEMENTOS DEL LENGUAJE DE PROGRAMACION

UNIDAD

I I

ESTRUCTURAS DE CONTROL

UNIDAD

I I I

ESTRUCTURAS DE DATOS

UNIDAD

I V

FUNCIONES