AyP1

Introducción a la programación

Explicación Práctica 1 - Parte 1

¿Cómo se estructura un programa en Pascal?

Pascal

```
program ejemploPascal;
var
  {variables del programa}
begin
  {cuerpo del programa}
end.
```

¿Qué tipos de variables existen en Pascal?

Pascal

Tipo de variable	Datos
Integer	Números enteros
Real	Números reales
Boolean	True – False
Char	Caracteres
Otros ...	

¿Cómo se declaran variables en Pascal?

Pascal

```
program ejemploPascal;
var
  nombre_variable: tipo
begin
  {cuerpo del programa}
end.
```

¿Cómo se da valor a una variable?

Pascal

- Usando el **operador** `:=`
- Mediante la **operación de lectura** de teclado: `read(variable)`

¿Cómo se imprime el valor de una variable?

Pascal

Mediante la **operación de escritura** de pantalla:

```
write(variable)
```

Veamos un ejemplo en Pascal

Implementar un programa en ***Pascal*** que lea de teclado dos números enteros, realice la suma de los mismos e imprima en pantalla el resultado obtenido.

```
program ejercicio;

var
  num1, num2, suma: integer;

begin
  read(num1);
  read(num2);
  suma := num1 + num2;
  write('El resultado es: ', suma);
end.
```

PARA PENSAR

- ¿Qué modificaciones deberían hacerse en el programa si se quisiera informar, además del resultado obtenido, los números que fueron sumados?

- ¿Y si se quisiera informar el doble del resultado obtenido?

FACULTAD DE INFORMATICA
UNIVERSIDAD NACIONAL DE LA PLATA

Introducción a la programación

Explicación Práctica 1 - Parte 2

Estructura de control: Decisión

*La **Decisión** permite cambiar el flujo de control del algoritmo dependiendo si una condición es verdadera o falsa.*

La condición debe poder tomar un valor binario: Verdadero o Falso

Sintaxis:

```
if  (condición) then
        Conjunto de instrucciones a realizar si la condición es verdadera
Else
        Conjunto de instrucciones a realizar si la condición es falsa
```

Opcional

Estructura de control: Decisión

```
if (condición) then
  accion;
```

más de una acción

```
if (condición) then
 begin
  acción 1;
  acción 2;
 end;
```

```
if (condición) then
  acción 1
else
  acción 2;
```

```
if (condición) then
 begin
  acción 1;
  acción 2;
 end
else
  acción 3;
```

```
if (condición) then
 begin
  acción 1;
  acción 2;
 end
else
  begin
   acción 3;
   acción 4;
  end;
```

EJEMPLOS DE USO

Estructura de control: `if`

Realice un programa que lea de teclado dos números enteros e informe el resultado de la suma de ambos, **sólo si éste es mayor que 50**.

```
program sumaMayor50;
var
  numero1, numero2, res: integer;
begin
  readln(numero1);
  readln(numero2);
  res := numero1 + numero2;
  if (res > 50)then
    writeln('El resultado es: ', res);
end.
```

EJEMPLOS DE USO

Estructura de control: `if - else`

Realice un programa que lea de teclado un número entero que representa la nota de un examen final e informe **si el alumno aprobó o no**. Considere que este examen se aprueba con 4 o más.

```
program notaExamen;
var
  nota: integer;
begin
  readln(nota);
  if (nota >= 4)then
    writeln('El alumno aprobó')
  else
    writeln('El alumno no aprobó');
end.
```

AyP1

Introducción a la programación

Explicación Práctica 1 - Parte 3

Lógica proposicional

*Una proposición es una expresión de la cual tiene sentido decir si es **Verdadera** o **Falsa**, o sea que es posible asignarle un valor de verdad (Verdadero o Falso, pero no ambos).*

Ejemplos:

La Plata es una provincia

1 + 1 = 2

Hay papel en la bolsa

Lógica proposicional

Clasificación

Proposiciones **atómicas**	Pueden ser representadas por una variable lógica y no pueden ser subdivididas
Proposiciones **moleculares**	Son un conjunto de proposiciones atómicas relacionadas con **operadores lógicos**. Pueden ser subdivididas.

Lógica proposicional

Operadores lógicos

Operador	Simbolización matemática	Ejemplo
Conjunción (y)	^	Llueve **y** hace frío
Disyunción (o)	v	Es Lunes **o** es martes
Negación (no)	¬	**No** hay comida

Lógica proposicional

Proposiciones. Ejemplos

Mi perro es verde ⇢ Atómica

La casa es grande ⇢ Atómica

Hoy es viernes y hay teoría ⇢ Molecular ¿Operador? y

No Hay una flor en la esquina ⇢ Molecular ¿Operador? No

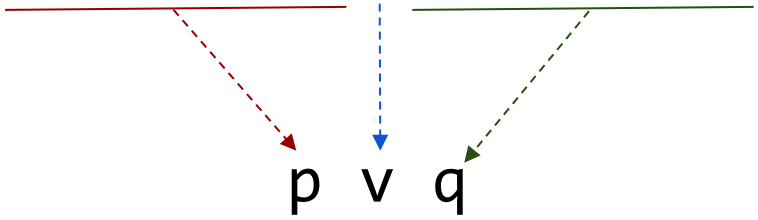
La casa es grande o el mate está lavado ⇢ Molecular ¿Operador? O

Lógica proposicional

Simbolización de proposiciones

*Las proposiciones suelen simbolizarse con letras minúsculas como **p**, **q**, **r**, etc.*

Por ejemplo: La casa es grande o el mate está lavado



Lógica proposicional

Simbolización de proposiciones. Ejemplos

- “Juan mide más de dos metros y no es jugador de básquet”

Simbolización: **p** = Juan mide más de dos metros
q = Juan es jugador de básquet

p ^ ~q

- “El cielo no es azul o América no es un océano"

Simbolización: **p** = El cielo es azul
q = América es un océano

~p v ~q

Lógica proposicional

Tablas de verdad

*Las **tablas de verdad** muestran el valor de verdad de una **proposición molecular** para cada combinación de verdad que es posible asignar a sus proposiciones atómicas.*

Lógica proposicional

Tablas de verdad

Conjunción

p	q	p ^ q
V	V	V
V	F	F
F	V	F
F	F	F

Sólo en el caso en que ambas proposiciones sean V la conjunción será V.

Disyunción

p	q	p v q
V	V	V
V	F	V
F	V	V
F	F	F

Sólo en el caso en que ambas proposiciones sean F, la disyunción será F.

Negación

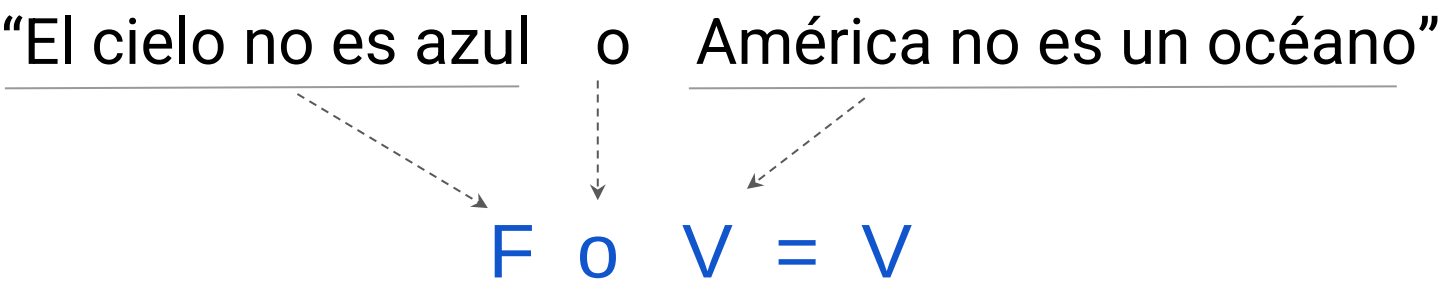
p	¬p
V	F
F	V

Se invierte el valor de la proposición

Lógica proposicional

Tablas de verdad.

Ejemplo de asignación de valores de verdad



Ejemplos

Leer un número entero e informar si es par y mayor que 10.

```
Program ejemplo;
Var
        num: integer;
Begin
    readln(num);
    if (num mod 2 = 0 ) and (num > 10) then
        Write('El número es par y mayor que 10');
end.
```

Para informar, las dos condiciones deben ser verdaderas

Leer un carácter e informar si es una vocal.

```
Program ejemplo;
Var
        car: char;
Begin
    readln(car);
    if (car ='a') or (car ='e') or (car ='i') or (car ='o') or  (car ='u')then
        Write('El carácter es una vocal');
end.
```

Para informar al menos una de las condiciones debe ser verdadera

FACULTAD DE INFORMATICA
UNIVERSIDAD NACIONAL DE LA PLATA

Introducción a la programación

Explicación Práctica 1 - Parte 4

Estructura de control: Iteración

*La **Iteración** permite repetir un grupo de instrucciones hasta alcanzar una condición. Se utiliza cuando se desconoce el número de repeticiones necesario.*

Sintaxis:

```
while (condición) do
    Conjunto de instrucciones a ejecutar
```

La condición debe poder tomar un valor binario: Verdadero o Falso

Estructura de control: Iteración

```
while (condición) do
  accion;
```

más de
una acción

```
while(condición) do
 begin
  acción 1;
  acción 2;
 end;
```

EJEMPLOS DE USO

Estructura de control: `while`

Realice un programa que lea de teclado números enteros **hasta que se ingrese el 0 (cero)** e informe la cantidad de números leídos.

```
Program numeros;
Var
  numero, cant: integer;
Begin
  cant:= 0;
  read(numero);
  while (numero <> 0) do begin
    cant:= cant +1;
    read(numero);
  end;
  writeln('La cantidad de números leídos es: ',cant);
End.
```

EJERCICIO

Realizar un programa que lea de teclado números enteros **hasta que se ingrese el 0 (cero)** e informe la cantidad de **números menores que 100**.

```
Program ejercicioExp1;
var
  numero, cant: integer;
begin
  cant := 0;
  read(numero);
  while(numero <> 0) do begin
    if(numero < 100) then
      cant:= cant + 1;
    read(numero);
  end;
  write('La cantidad de números menores que 100 es', cant);
end.
```

Estructura de control: Repetición

*La **Repetición** permite repetir un número fijo de veces un grupo de instrucciones*

Sintaxis:

```
For i:= valor_inicial to valor_final do
    Conjunto de instrucciones a repetir
```

Estructura de control: Repetición

```
for indice :=  valor_inicial to valor_final do
    accion 1;
```

más de
una acción

```
for indice :=  valor_inicial to valor_final do
 begin
    accion 1;
     accion 2;
 end;
```

EJEMPLOS DE USO

Estructura de control: `for`

Realice un programa que lea de teclado 10 números enteros e informe el resultado de la suma.

```
Program suma;
Var
  i, numero, res: integer;
Begin
  res := 0;
  for i:= 1 to 10 do begin
    readln(numero);
    res:= res + numero;
  end;
  writeln('La suma es:', res);
End.
```

PARA RESOLVER

Estructura de control: `for`

¿Qué imprime el siguiente código?

```
program queImprime;
var
  i: integer;
begin
  for i:= 1 to 5 do
    writeln(i);
end.
```

El índice de un for no debe modificarse. ¿Qué pasa si ejecutamos el siguiente código?

```
program infinito;
var
  i: integer;
begin
  for i:= 1 to 5 do begin
    writeln(i);
    i:= 1;
  end;
  readln();
end.
```

PARA RESOLVER

Estructura de control: `for`

¿Qué imprime el siguiente código?

```
Program queImprime2;
Var
  i: integer;
Begin
  for i:= 1 to 5 do
    if ((i mod 2) = 0) then
      writeln(i);
End.
```

AyP1

Introducción a la programación

Explicación Práctica 1 - Parte 5

Máximos y Mínimos

CALCULAR VALOR MÁXIMO

Realizar un programa que lea números naturales desde teclado. La lectura debe finalizar cuando se ingrese el número 0, el cual **no** debe procesarse.

Informar en pantalla cuál es el número máximo leído.

¿Estructura de control?

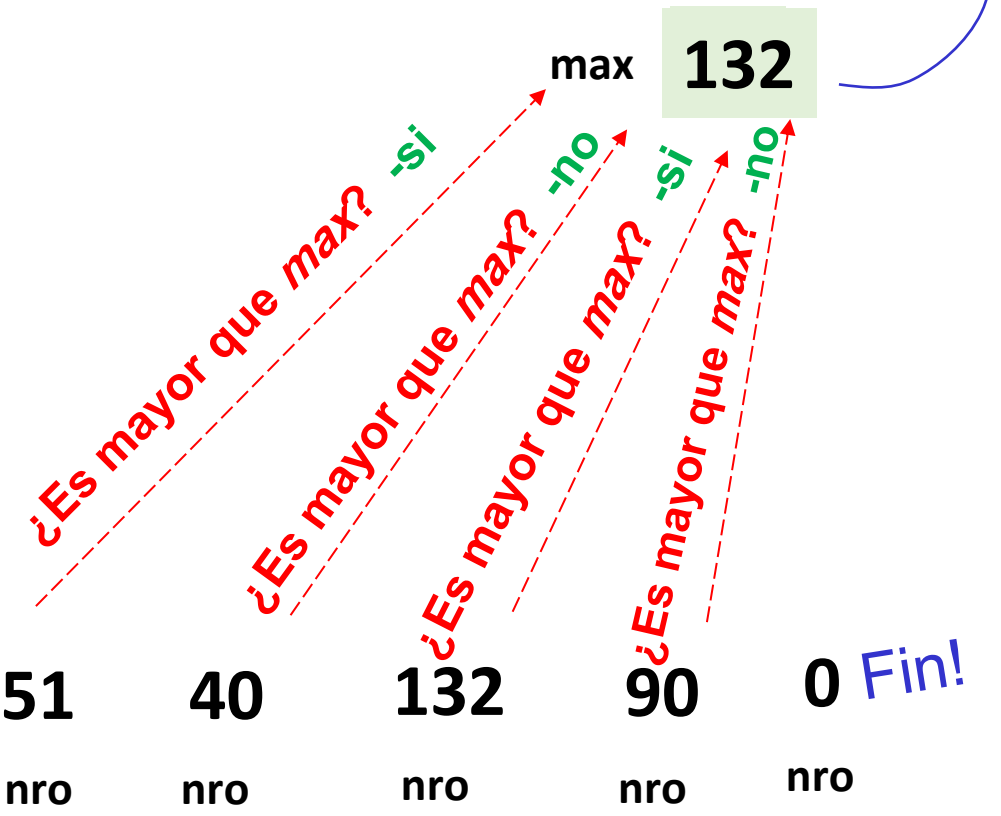
¿Datos a calcular?

¿Datos a leer de teclado ?

CALCULAR VALOR MÁXIMO

Analizando el problema…

Variable para llevar el máximo, inicializada con un valor muy bajo.



Se comienza a leer hasta que llegue el 0…

El número leído más alto fue: 132

CALCULAR VALOR MÁXIMO

Realizar un programa que lea números naturales desde teclado. La lectura debe finalizar cuando se ingrese el número 0, el cual **no** debe procesarse.

Informar en pantalla cuál es el número máximo leído.

```
program valorMaximo;
var
  nro, max: integer;
Begin
  max:= -1;
  readln(nro); {leo un número}
  while(nro <> 0) do begin
    if (nro > max) then  {evalua el máximo}
      max:= nro;
    readln(nro); {leo otro número}
  end;
  writeln('El número más alto fue: ', max);
end.
```

CALCULAR VALOR MÁXIMO

Realizar un programa que lea números naturales desde teclado. La lectura debe finalizar cuando se ingrese el número 0, el cual **no** debe procesarse.

Informar en pantalla cuáles son los 2 número máximos leídos.

¿Estructura de control?

¿Datos a calcular?

¿Datos a leer de teclado ?

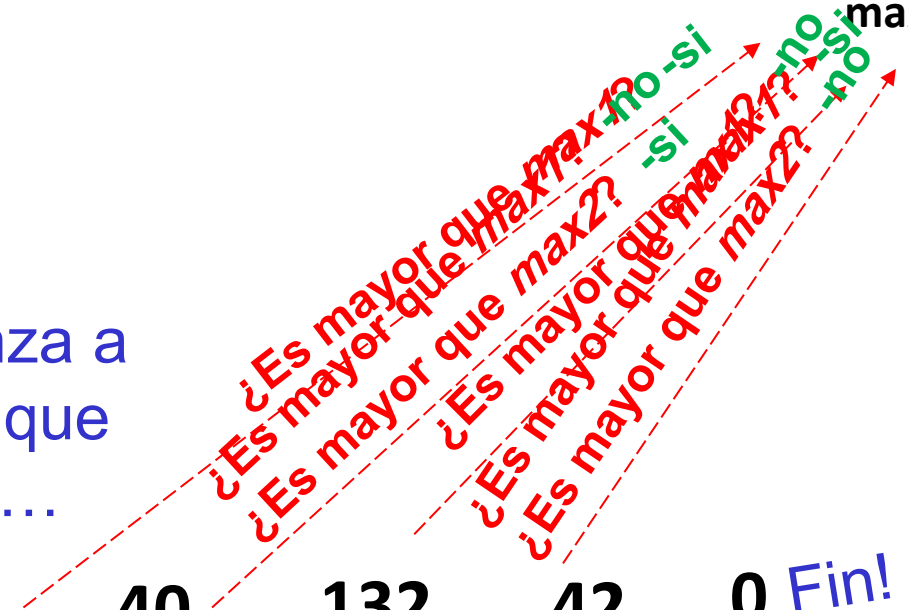
CALCULAR 2 VALORES MÁXIMOS

Analizando el problema…

Variables para llevar el máximo1 y el máximo2.

max1 **132**

max2 **51**



Se comienza a leer hasta que llegue el 0…

51 nro **40** nro **132** nro **42** nro **0** nro Fin!

Los 2 números más alto fueron:
132 y 51

CALCULAR 2 MÁXIMOS

```
program DosMaximos;
var
  max1, max2: integer;
  n: integer;
begin
  max1:=-1; max2:=-1; {inicializa los maximos}
  read(n);
  while (n <> 0) do begin
    if (n > max1) then begin {evalua máximo 1}
      max2:=max1;
      max1:=n;
    end
    else
      if (n > max2) then {evalua máximo 2}
        max2:=n;
    read(n);
  end;
  writeln('Los 2 números mas altos fueron', max1,'y', max2);
end.
```