

Ayuda memoria

Estructura de un programa

```
Programa Nombre
  Comenzar
      .....

  Fin
```

Leer e imprimir una variable

```
Programa Ejemplo
  Comenzar
      leer (a)
      imprimir(a)

  Fin
  Es equivalente imprimir a escribir.
```

Inicializar una variable

```
Programa Ejemplo
  Comenzar
      a := 1

  Fin
```

Sumar 1 a una variable

```
Programa Ejemplo
  Comenzar
      a := 1
      a := a + 1
      imprimir(2)

  Fin
  El valor de a seria 2
```

Leer un numero e imprimir si es par

```
Programa Ejemplo
  Comenzar
      leer(a)
      si (a mod 2 == 0) entonces
          imprimir("es par")
      sino
          imprimir("es impar")

  Fin
  Mod es una función que retorna el resto entre
  dos números
```

Repetir 10 veces e imprimir

```
Programa Ejemplo
  Comenzar
      repetir i:=0 hasta 10 hacer
          imprimir(0)
      fin repetir

  fin
  Imprime los numeros de 0 hasta 10,
  recuerden que el valor de la variable i se
  incrementa automaticamente.
```

Leer n numeros hasta que ingrese 999 e imprimirlos

```
Programa Ejemplo
  Comenzar
      leer (a)
      mientras(a<>999)hacer
          imprimir(a)
          leer(a)
      fin repetir

  Fin
```

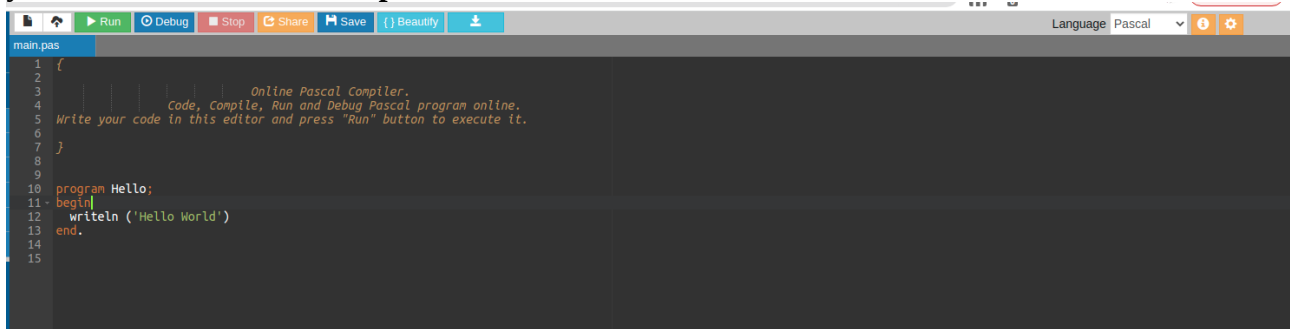
Leer n numeros hasta que ingrese 999 e imprimir el mayor

```
Programa Ejemplo
  Comenzar
      leer (a)
      aux_mayor :=a
      mientras(a<>999) hacer
          si (a > aux_mayor) entonces
              aux_mayor := a
          leer(a)
      fin repetir
      imprimir("el mayor es")
      imprimir(a)

  Fin
```

Ayuda memoria 2 – Pascal

Se puede utilizar: https://www.onlinegdb.com/online_pascal_compiler o cualquier IDE. Lo bueno de los IDEs es que tiene debug. Free Pascal es la interfaz carácter, y lazarus la interfaz mas copada.



Se pueden fijar arriba en lenguaje indicar pascal.

Repetir 10 veces e imprimir

```
program Ejemplo;
  var i:integer;
begin
  for i:= 0 to 10 do
    begin
      Write(i)
    end
  end.
end.
```

Sumar 1 a una variable

```
program Ejemplo;
  var i:integer;
begin
  read(i);
  i:= i+1;
  Write(i);
end.
```

Leer un numero e imprimir si es par

```
program Ejemplo;
  var i:integer;
begin
  read(i);
  if (i mod 2 = 0)then
    Write('es par')
  else
    Write('es impar')
end.
```

Leer n numeros hasta que ingrese 999 e imprimirlos

```
program Ejemplo;
  var i:integer;
begin
  read(i);
  while(i<>999)do
    Begin
      write(i)
      read(i)
    end
  end.
end.
```

Leer n numeros hasta que ingrese 999 e imprimir el mayor

```
program Ejemplo;
  var i,mayor:integer;
begin
  read(i);
  mayor:=i;
  while(i <> 999) do
    begin
      if(i>mayor)then
        mayor := i;
      read(i);
    end;
    Write('El mayor es',mayor);
  end.
```

Leer n numeros hasta que ingrese 999 e imprimir el mayor y menor

```
program Ejemplo;
  var i,mayor,menor:integer;
begin
  read(i);
  mayor := i;
  menor := i;
  while(i<>999)do
    begin
      if(i>mayor)then
        begin
          mayor := i;
        end
      else if(i<menor)then
        begin
          menor := i;
        end;
      read(i)
    end;
    writeln('el menor es ',menor);
    writeln('el mayor es ',mayor);
  end.
```

Pasos a tener en cuenta para realizar un algoritmo:

- 1- Detectar variables
- 2- Detectar que estructura de control debo utilizar.
- 3- Declarar variables
- 4- Inicializar variables
- 5- Leer de consola (si es necesario)
- 6- Escribir la estructura de control (for o repetir)
- 7- Realizar las tareas dentro de la estructura de control, sumar, contar, etc.
- 8- Volver a leer de consola (si es necesario, para que la estructura itere)
- 9- finalizar la estructura de control.
- 10- Informar resultados.

Ejemplo:

Leer n números hasta ingresar 999, contar los números pares, informar estos e informar la cantidad total de números leídos.

```
program Ejemplo;
var n,pares,total:integer; {punto 1 y 3}
begin
  pares := 0; {punto 1 y 4}
  total := 0;
  read(n); {punto 5}
  while(n<>999)do {punto 2 y 6}
  begin
    if (n mod 2 = 0)then {punto 7}
    begin
      pares := pares +1;
    end;
    total := total +1;
    read(n); {punto 8}
  end; {punto 9}
  writeln ('Cantidad de pares ', pares); {punto 10}
  writeln ('Cantidad total ', total); {punto 10}
end.
```

Arreglos

Crear un array de elementos random

```
program Ejemplo;
type arreglo= array [1..50] of integer;
var a:arreglo;
    i:integer;
begin
    for i:=1 to 50 do
        a[i] := Random(100);
    end.
```

Imprimir un array

```
procedure imprimir(a:arreglo;longitud:integer);
var i :integer;
begin
    for i:=1 to longitud do
        writeln(a[i] );
    end;
```

Buscar un elemento en un array

```
function
buscar(a:arreglo;longitud:integer;elementoABuscar:integer):boolean;
var i :integer;
    encontro:boolean;
begin
    encontro:=false;
    for i:=1 to longitud do
        if (a[i] = elementoABuscar)then
            encontro:= true;
    buscar:=encontro;
end;
```

Crear, cargar e imprimir una matriz

```
program Ejemplo;
type filas= array [1..5] of integer;
    matriz=array[1..5] of filas;
var m:matriz;
    i,j:integer;

procedure imprimir(a:matriz;filas:integer;columnas:integer);
var p,r:integer;
begin
    for p:= 1 to columnas do
        begin
            for r:= 1 to filas do
                Write(a[p][r], ' ');
                Writeln("");
            end;
        end;
end;

begin
    for i:= 1 to 5 do
        for j:= 1 to 5 do
            m[i][j]:=Random(100);
        imprimir(m,5,5)
    end.
```

Registros

Crear un registro alumno

```
program Ejemplo;  
type alumno = record  
  nombre:string;  
  apellido:string;  
end;  
var a:alumno;  
begin  
  a.nombre := 'Test 1' ;  
  a.apellido := 'Test 2' ;  
  writeln(a.nombre);  
  writeln(a.apellido);  
end.
```

Array de alumnos

```
program Ejemplo;  
type alumno = record  
  nombre:string;  
  apellido:string;  
end;  
  
var a: array [1..100] of alumno;  
begin  
  a[1].nombre := 'Test 1' ;  
  a[1].apellido := 'Test 2' ;  
  writeln(a[1].nombre);  
  writeln(a[1].apellido);  
end.
```

Alumnos y sus notas

```
type alumno = record  
  nombre:string;  
  apellido:string;  
  notas:array[1..3]of integer;  
end;
```