

Resolución de Problemas y Algoritmos

Uso de iteración con secuencias

Secuencias

En el procesamiento de secuencias de valores, éstos pueden ser leídos por consola (utilizando el buffer), o a través de algún otro dispositivo, como veremos más adelante.

Quando los valores se ingresan por consola es posible:

- Conocer la cantidad de valores antes de comenzar a leerlos.
- Utilizar un valor específico para indicar fin de la secuencia.
- Preguntarle al usuario luego de leer cada valor, si desea ingresar otro.

Secuencias

- Conocer la cantidad de valores antes de comenzar a leerlos

Ingrese la cantidad de valores: 5
Ingrese los valores: 23 4 7 12 9

- Utilizar un valor específico para indicar fin de la secuencia.

Ingrese los valores, para finalizar ingrese 0
23 4 7 12 9 6 31 0

- Preguntarle al usuario luego de leer cada valor, si desea ingresar otro.

Ingrese los valores:
23
Desea Ingresar otro (S/N):S
4
Desea Ingresar otro (S/N):S
7
.....

Secuencias

Ejercicio:

Escribir un programa que lea una secuencia de n valores reales y muestre cada uno de ellos seguido de su cuadrado.

En este caso antes de comenzar a procesar la secuencia, se conoce el número de valores que van a leerse y procesarse.

Algoritmo Cuadrados

DE: n , secuencia de números

Para i desde 1 hasta n
leer($elem$)
mostrar ($elemento$, $cuadrado$)

```
for i := 1 to n do
begin
  read (x);
  writeln (x,sqr(x));
end
```

La secuencia de valores ingresada por el usuario se mantiene en un área de memoria llamada **buffer**. La instrucción **read** consume el buffer.

Secuencias

```
program cuadrados;
{Muestra el cuadrado de cada número leído}
var i,n : integer;
    x : real;
begin
  write ('Ingrese el valor de n ');
  readln (n);
  for i := 1 to n do
  begin
    read (x);
    writeln (x,sqr(x));
  end;
end.
```

Secuencias

Ejercicio:

Escriba un programa que lea una secuencia de números enteros terminada con un 0 y muestre los que son múltiplos de n y mayores que 0.

En este caso no conocemos antes de comenzar a iterar la cantidad de valores que forman la secuencia. Un valor específico termina la secuencia, sin formar parte de ella.

Algoritmo Múltiplos

DE: secuencia de números, n

repetir
leer(s)
si ($s > 0$) y (s múltiplo de n)
entonces
mostrar (s)
hasta $s=0$

```
repeat
  read(s);
  if (s>0) and (s mod n = 0)
  then
    write (s,' ');
until (s = 0);
```

Secuencias

Ejercicio:

Escriba un programa para que lea por consola números enteros y cuente cuántos números de exactamente dos dígitos se ingresaron. El usuario decidirá cuando finalizar el ingreso de datos.

```

Algoritmo DosDígito
DE:secuencia de números, opcion
DS: contador

contador ← 0
opcion ← si
Mientras (opcion=si)
hacer
  leer(n)
  si (n)>9) y (n)<100)
  entonces
    contador ← contador + 1
  leer (opcion)

```

```

contador:=0;
writeln('Desea ingresar (S-N) ');
read(opcion);
while(opcion='S') do
begin
  read (n);
  if (ABS(n)>9) and (ABS(n)<100)
  then
    contador:= contador + 1;
  writeln('Desea ingresar (S-N) ');
  read(opcion);
End;

```

Secuencias

Ejercicio:

Escriba un programa que lea una secuencia de números positivos terminada en 0 y devuelva el mayor.

Por ejemplo, si la secuencia es 3 11 5 8 8 11 2 19 13 0
El mayor es 19

Secuencias

El usuario ingresa la secuencia completa.

3 11 5 8 8 11 2 19 13 0

El programa lee 3
Por el momento es el mayor

Secuencias

El usuario ingresa la secuencia completa.

3 11 5 8 8 11 2 19 13 0

El programa lee 3
Por el momento es el mayor

El programa lee el 11
Se compara el 11 con el 3

Secuencias

El usuario ingresa la secuencia completa.

3 11 5 8 8 11 2 19 13 0

El programa lee 3
Por el momento es el mayor

El programa lee el 11
Se compara el 11 con el 3
El 11 es el mayor

Secuencias

El usuario ingresa la secuencia completa.

3 11 5 8 8 11 2 19 13 0

El programa lee 3
Por el momento es el mayor

El programa lee el 11
Se compara el 11 con el 3
El 11 es el mayor

El programa lee el 5
Se compara el 5 con el 11
El mayor sigue siendo el 11

Secuencias

3 11 5 8 8 11 2 19 13 0

```
read (numero);
mayor := numero;
```

Necesitamos **dos** variables: numero y mayor

El programa lee un valor.
Por el momento es el mayor.

Secuencias

3 11 5 8 8 11 2 19 13 0

```
read (numero);
mayor := numero;
```

```
read (numero);
if (numero > mayor) then
    mayor := numero;
```

El programa lee el segundo valor.
Se compara el mayor con el último leído.
El segundo es el mayor

Secuencias

3 11 5 8 8 11 2 19 13 0

```
read (numero);
mayor := numero;
```

```
read (numero);
if (numero > mayor) then
    mayor := numero;
```

```
read (numero);
if (numero > mayor) then
    mayor := numero;
```

El programa lee el tercer valor.
Se compara el mayor con el último leído, si el último es mayor se actualiza.

Secuencias

3 11 5 8 8 11 2 19 13 0

```
read (numero);
mayor := numero;
```

```
read (numero);
if (numero > mayor) then
    mayor := numero;
```

```
read (numero);
if (numero > mayor) then
    mayor := numero;
```

Secuencias

3 11 5 8 8 11 2 19 13 0

```
read (numero);
mayor := numero;
```

```
While ( numero <> 0 ) do
begin
```

```
    read (numero);
    if (numero > mayor) then
        mayor := numero;
```

```
end;
```

Qué pasará si la secuencia es 0?

Secuencias

Ejercicio:

Escriba un programa que lea una secuencia de números positivos terminada en 0 y devuelva la posición del mayor.

Por ejemplo, dada la secuencia: 3 11 5 8 8 11 2 19 13 0
La posición del mayor es 8

Necesitamos **cuatro** variables: numero, mayor, pos y posMayor.

Secuencias

3 11 5 8 8 11 2 19 13 0

```
leer(numero)
mayor ← numero
pos ← 1
posMayor ← 1
```

```
Mientras ( numero <> 0 ) hacer
  leer(numero)
  pos ← pos + 1
```

```
Si (numero > mayor) entonces
  mayor ← numero
  posMayor ← pos
```