

Tecnologías en la Educación Matemática

Secuencias

Dpto. de Ciencias e Ingeniería de la Computación
UNIVERSIDAD NACIONAL DEL SUR



Procesamiento de Secuencias



Un tipo particular de problemas para el cual resulta adecuada la iteración es el procesamiento de **secuencias de valores**.

Los valores pueden ser leídos por consola o desde algún otro dispositivo (archivos por ejemplo, no se ven en esta materia).

Cuando 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.

2

Procesamiento de Secuencias

Problema: Leer una secuencia de **n** valores y calcular la suma de todos los valores.



Consumir toda la secuencia de **n** elementos.

```

Algoritmo sumaSec
DE: n (secuencia de n valores)
DS: suma
suma ← 0
para i desde 1 hasta n
  leer(v)
  suma ← suma+v
  
```

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

3

Procesamiento de Secuencias

Problema: Leer una secuencia de **n** valores y calcular la suma de todos los valores.



```

program SumaDeCuadrados;
var i, n, v, suma: integer;
begin
  write ('Ingrese la long. de la secuencia:'); readln (n);
  write ('INGRESE LA SECUENCIA:'); suma:=0;
  for i := 1 to n do
    begin
      read(v);
      suma := suma + v;
    end;
  Writeln('La suma es:', suma);
end.
  
```

4

Procesamiento de Secuencias

Problema: Leer una secuencia de **N** números enteros y calcular la cantidad de veces que aparece el entero **M**.

N y **M** son valores ingresados por el usuario inicialmente.

Casos de prueba:

N=5 M= 10 y la secuencia es 1 2 3 10 2

N=8 M=-1 y la secuencia es -1 2 3 -1 -1 2 3 -1

N=4 M=4 y la secuencia es 2 5 6 2

¿Cuáles son los datos de entrada?

¿Cuáles son los datos de salida?

¿Conocemos la cantidad de iteraciones?

5

Procesamiento de Secuencias



```

program contarEnSecuencia;
{Lee una secuencia de N elementos enteros y cuenta cuantos son iguales a M con N y M leídos previamente}
var N, M, elem, i, cont: integer;
begin
  ...
end.
  
```

6

Procesamiento de Secuencias



```
...
write ('Ingrese la longitud de la secuencia:'); readln (N);
write ('Ingrese el elemento a buscar:'); readln (M);
cont := 0;
write ('Ingrese una secuencia de ', N, ' enteros');
for i:=1 to N
do
begin
read(elem);
if (elem=M)
then cont := cont+1;
end;
writeln (M, ' está ', cont, ' veces ');
...
```

7

Procesamiento de Secuencias



Problema: Leer una secuencia de N números enteros y determinar si el entero M aparece al menos una vez.

N y M son valores ingresados por el usuario inicialmente.

Casos de prueba:

N=5 M= 10 y la secuencia es 1 2 3 10 2

N=8 M=-1 y la secuencia es -1 2 3 -1 -1 2 3 -1

N=4 M=4 y la secuencia es 2 5 6 2

Conozco la longitud de la secuencia, uso un *for*??

8

Procesamiento de Secuencias



```
write ('Ingrese la longitud de la secuencia:'); readln (N);
write ('Ingrese el elemento a detectar:'); readln (M);
esta := false;
```

```
write ('Ingrese la secuencia de ', n, ' enteros');
```

```
for i:=1 to N
```

```
do
```

```
begin
```

```
read (elem);
```

```
if (elem=M)
```

```
then esta:=true;
```

```
end;
```

En este caso, antes de comenzar a procesar la secuencia,

SE CONOCE el número de valores que van a leerse pero puede suceder que **NO NECESITEMOS** leerlos todos para conocer la solución.

Procesamiento de Secuencias



```
write ('Ingrese la longitud de la secuencia:'); readln (N);
```

```
write ('Ingrese el elemento a detectar:'); readln (M);
```

```
esta := false; i:=1;
```

```
write ('Ingrese la secuencia de ', n, ' enteros');
```

```
while (i<=N) and (esta=false)
```

```
do
```

```
begin
```

```
read (elem);
```

```
if (elem=M)
```

```
then esta:=true
```

```
else i:=i+1;
```

```
end;
```

```
...
```

Puede suceder que no necesitemos consumir toda la secuencia.

¡¡OPTIMIZAMOS!!

10

Procesamiento de Secuencias



Problema: Leer una secuencia de números enteros **terminada en 0** y **contar** cuántos son múltiplos de n, con n positivo e ingresado previamente por el usuario.

```
hayElem:=true;
```

```
contador :=0;
```

```
while (hayElem) do
```

```
begin
```

```
read(elem);
```

```
If (elem = 0)
```

```
then
```

```
hayElem:=false
```

```
else
```

```
if (elem mod n = 0) then
```

```
contador := contador + 1;
```

```
end;
```

En este caso, antes de comenzar a procesar la secuencia,

NO SE CONOCE el número de valores que van a leerse.

Mientras que haya elementos en la secuencia, leo un elemento y si es múltiplo de n incremento el contador.

Consumir toda la secuencia hasta llegar al 0.

11

Procesamiento de Secuencias



```
program ContarMultiplosSecuencia;
```

```
var elem, contador, n: integer; hayElem: boolean;
```

```
begin
```

```
write ('Ingrese la longitud de la secuencia: '); readln (n);
```

```
writeln('Ingrese la secuencia: ');
```

```
hayElem:=true; contador:=0;
```

```
while (hayElem)
```

```
do
```

```
begin
```

```
read(elem);
```

```
If (elem = 0)
```

```
then hayElem:=false
```

```
else if (elem mod n = 0) then
```

```
contador := contador + 1;
```

```
end;
```

```
writeln('Cantidad de multiplos: ', contador);
```

```
end.
```

12

Procesamiento de Secuencias



Problema: Leer una secuencia de números enteros **terminada en 0** y **determinar si hay algún múltiplo de n**, con n positivo e ingresado previamente por el usuario.

```

hayElem:=true;  encuentre:=false;
while (hayElem) and (not encuentre) do
  begin
    read(elem);
    if (elem = 0)
    then
      hayElem:=false
    else
      if (elem mod n = 0) then
        encuentre:=true;
  end;
end;
    
```

Puede no ser necesario consumir toda la secuencia hasta llegar al 0.
Condición múltiple



Procesamiento de Secuencias



Problema: Leer una secuencia de números positivos terminada en 0 y obtener el mayor.

Por ejemplo, dada la secuencia

3 11 5 8 8 11 2 19 13 0

El mayor es 19



Procesamiento de Secuencias



```

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

...



Procesamiento de Secuencias



```

program maximo;
var numero, mayor: integer;
begin
  writeln('Ingrese una secuencia finalizando en 0');
  read (numero);
  mayor := numero;
  while (numero <> 0) do
    begin
      read (numero);
      if (numero > mayor)
      then mayor := numero;
    end;
  writeln('El mayor de la secuencia es:', mayor);
end.
    
```

Considere la secuencia VACÍA



Procesamiento de Secuencias



Preguntas que deberíamos hacernos antes de diseñar el algoritmo:

- La longitud de la secuencia, ¿es conocida?
- ¿Es necesario analizar todos los elementos de la secuencia para conocer el resultado?

Si no es necesario, optimizamos!!



Procesamiento de Secuencias



		Conozco la longitud de la secuencia?	
		SI	NO
Necesito recorrerla completa?	SI	for i:=1 to long do ...	fin:=false; while (not fin) do ...
	NO	i:=1; encuentre:=false; while (i<=long) and (not encuentre) do begin i:=i+1; end;	fin:=false; encuentre:=false; while (not fin) and (not encuentre) do



Procesamiento de Secuencias



Problema: Leer una secuencia de números positivos terminada en 0 (que se considera elemento de la misma) y obtener el mayor y el menor elementos de la secuencia. Implemente un procedimiento para resolver este problema.

Por ejemplo, dada la secuencia

3 11 5 8 8 11 2 19 13 0

El mayor es el 19 y el menor es el 2.



Procesamiento de Secuencias



Problema: Leer una secuencia de números positivos terminada en 0 y obtener la posición del mayor. Implemente una función para resolver este problema.

Por ejemplo, dada la secuencia

3 11 5 8 8 11 2 19 13 0

La posición del mayor es 8



Procesamiento de Secuencias



Suponga una **secuencia de caracteres terminada en punto** ingresada por teclado. Desarrolle un programa en Pascal que informe la suma de los dígitos que aparecen en la secuencia.

Por ejemplo, sea la secuencia: "El número 528 tiene 3 dígitos; el número 17 tiene solo 2 dígitos." el programa deberá mostrar por pantalla: "La suma de los dígitos presentes en esta secuencia es 28". (Ejercicio del práctico)

Importante!

- ❖ La secuencia está compuesta por caracteres.
- ❖ Cada dígito también se lee como un carácter.
- ❖ Luego lo transformamos a dígito numérico.

21

Procesamiento de Secuencias



Primera aproximación:

Mientras que no se termine la secuencia
leer un carácter
si es un dígito convertirlo a número
acumularlo

De carácter a dígito (para los caracteres del '0' al '9')

ord('0') = 48
ord('0') - 48 = 0

ord('1') = 49
ord('1') - 48 = 1

ord('2') = 50
ord('2') - 48 = 2

25	49	73
26	50	74
27	51	75
28	52	76
29	53	77
30	54	78
31	55	79
32	56	80
33	57	81
34	58	82
35	59	83
36	60	84
37	61	85
38	62	86
39	63	87
40	64	88
41	65	89
42	66	90
43	67	91
44	68	92
45	69	93
46	70	94
47	71	95
48	72	96

22

Procesamiento de Secuencias

25	49	73
26	50	74
27	51	75
28	52	76
29	53	77
30	54	78
31	55	79
32	56	80
33	57	81
34	58	82
35	59	83
36	60	84
37	61	85
38	62	86
39	63	87
40	64	88
41	65	89
42	66	90
43	67	91
44	68	92
45	69	93
46	70	94
47	71	95
48	72	96

```

program ejPracticoSecuencias;
var ch: char; d, suma: integer;
    fin: boolean;
begin
  Write('Ingrese el texto: ');
  fin:=false; suma:=0;
  while (fin=false) do
  begin
    read(ch);
    if (ch='.')
    then fin:=true
    else
      if (ch>='0') and (ch<='9')
      then
        begin
          d:= ord(ch)-48;
          suma:=suma + d;
        end;
    end;
  end;
  Write('La suma de los digitos es: ', suma);
  readln; readln;
end.
    
```

23