

VECTORES CONTADORES

EXPLICACIÓN PRÁCTICA 4 continuación

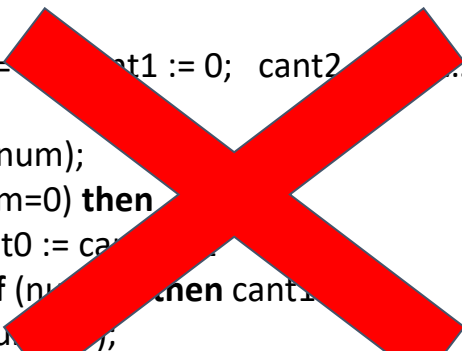
CADP 2023

Ejemplo

Se lee una secuencia de dígitos (números entre el 0 y el 9) hasta que se ingresa el cero, que debe procesarse. Al finalizar la secuencia, informar la cantidad de veces que aparece cada dígito.

Secuencia de prueba: 4 6 9 8 4 9 5 7 1 9 0

```
var
  num; cant0, cant1, cant2, cant3, ... , cant9 : integer;
begin
  cant0 := 0; cant1 := 0; cant2 := 0; ...
  repeat
    read(num);
    if (num=0) then
      cant0 := cant0 + 1;
    else if (num=1) then cant1 := cant1 + 1;
    ...
  until (num = 0);
```



```
type
  vdigitos = array[0..9] of integer;
var
  digitos : vdigitos;
  num : integer;
begin
  inicializar(digitos);
  repeat
    read(num);
    digitos[num] := digitos[num] + 1;
  until (num = 0)
  imprimir(digitos);
end.
```

0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9

Ejercicio

Hacer un programa que lea una secuencia de números enteros terminada en 0. Informar la cantidad de veces que aparece cada dígito del 0 al 9 entre todos los números leídos.

Ejemplo: se leen los números: 457 9875 5 24879 0

{Se debe informar por cada dígito cuántas veces aparecieron cada uno de ellos}

0 aparece 0 veces;

1 aparece 0 veces;

2 aparece 1 veces;

3 aparece 0 veces;

4 aparece 2 veces;

5 aparece 3 veces;

6 aparece 0 veces;

7 aparece 3 veces;

8 aparece 2 veces;

9 aparece 2 veces;

¿Debo almacenar los números leídos?

¿Cómo cuento las veces que aparece cada dígito?

0	0	1	0	2	3	0	3	2	2
0	1	2	3	4	5	6	7	8	9

¿Necesito llevar la dimensión lógica?

```

Program Digitos;
Type
    rango=0..9;
    numeros=array [rango] of integer;

    {Acá se declaran Los módulos}
var
    losnros:numeros;
    num:integer;
begin
    inicializar(losnros);
    read(num);
    while (num <> 0) do begin
        descomponer(losnros, num);
        read(num);
    end;
    informo(losnros);
end.

```

```

procedure inicializar(var a:numeros);
var i:rango;
begin
    for i:=0 to 9 do
        a[i]:=0;
    end;

    procedure descomponer(var a:numeros;
        num:integer);
    var
        resto:rango;
    begin
        while (num <> 0) do begin
            resto:=num mod 10; {Obtengo digito}
            {Incremento contador asociado al digito}
            a[resto]:=a[resto] + 1;
            num:=num div 10; {Achico número}
        end;
    end;

    procedure informo(a:numeros);
    var
        i:rango;
    begin
        for i:=0 to 9 do
            writeln(i, ' = ',a[i]);
        end;
    end;

```

1. Modifique para informar para cada número la cantidad de veces que aparece cada dígito.

```
Program Digos;
Type
  rango=0..9;
  numeros=array [rango] of integer;

  {Acá se declaran los módulos}
var
  losnros:numeros;
  num:integer;
begin
  inicializar(losnros);
  read(num);
  while (num <> 0) do begin
    descomponer(losnros, num);
    read(num);
  end;
  informo(losnros);
end.
```

```
procedure inicializar(var a:numeros);
var i:rango;
begin
  for i:=0 to 9 do
    a[i]:=0;
  end;

procedure descomponer(var a:numeros;
  num:integer);
var
  resto:rango;
begin
  while (num <> 0) do begin
    resto:=num mod 10; {Obtengo dígito}
    {Incremento contador asociado al dígito}
    a[resto]:=a[resto] + 1;
    num:=num div 10; {Achico número}
  end;
end;

procedure informo(a:numeros);
var
  i:rango;
begin
  for i:=0 to 9 do
    writeln(i, ' = ', a[i]);
  end;
```

1. Modifique para informar para cada número la cantidad de veces que aparece cada dígito.

```
Program Digos;
Type
  rango=0..9;
  numeros=array [rango] of integer;

  {Acá se declaran los módulos}
var
  losnros:numeros;
  num:integer;
begin
  inicializar(losnros);
  read(num);
  while (num <> 0) do begin
    descomponer(losnros, num);
    read(num);
  end;
  informar(losnros);
end.
```

```
procedure inicializar(var a:numeros);
var i:rango;
begin
  for i:=0 to 9 do
    a[i]:=0;
  end;

procedure descomponer(var a:numeros;
  num:integer);
var
  resto:rango;
begin
  while (num <> 0) do begin
    resto:=num mod 10; {Obtengo dígito}
    {Incremento contador asociado al dígito}
    a[resto]:=a[resto] + 1;
    num:=num div 10; {Achico número}
  end;
end;

procedure informar(a:numeros);
var
  i:rango;
begin
  for i:=0 to 9 do
    writeln(i, ' = ', a[i]);
  end;
```

1. Modifique para informar para cada número la cantidad de veces que aparece cada dígito.

Program Digitos;

Type

rango=0..9;

numeros=array [rango] of integer;

{Acá se declaran Los módulos}

var

losnros:numeros;

num:integer;

begin

read(num);

while (num <> 0) do begin

inicializar(losnros);

descomponer(losnros, num);

informo(losnros);

read(num);

end;

end.

Program Digitos;

Type

rango=0..9;

numeros=array [rango] of integer;

{Acá se declaran Los módulos}

var

losnros:numeros;

num:integer;

begin

read(num);

while (num <> 0) **do begin**

inicializar(losnros);

descomponer(losnros, num);

informo(losnros);

read(num);

end;

end.

2. Modifique para informar
el dígito que **más veces**
apareció para cada número

Program Digitos;

Type

rango=0..9;

numeros=array [rango] of integer;

{Acá se declaran los módulos}

var

losnros: numeros;

num:integer;

begin

read(num);

while (num <> 0) **do begin**

inicializar(**losnros**);

descomponer(**losnros**, num);

write('el dig que mas aparece es', **DigitoMaximo**(**losnros**));

read(num);

end;

end.

Function **DigitoMaximo**(a:numeros):rango;

var

i, digmax:rango;

max:integer;

begin

max:=-1;

for i:=0 **to** 9 **do**

if (a[i] > max) **then begin**

max:=a[i];

digmax:=i;

end;

DigitoMaximo:=digmax;

end;

2. Modifique para informar el dígito que más veces apareció para cada número