

Ciclos anidados

ANIDAMIENTO DE CICLOS

- Consiste en usar un ciclo dentro de otro.

Bucles anidados

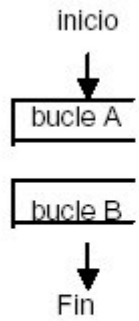
Los bucles anidados constan de un bucle externo con uno o más bucles internos. Cada vez que se repite el bucle externo, los bucles internos se repiten, se vuelven a evaluar los componentes de control y se ejecutan todas las iteraciones requeridas.

El **funcionamiento** de un ciclo anidado se lleva a cabo teniendo en cuenta que el ciclo más interno es el primero en ejecutar.

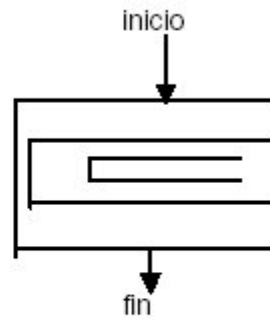
Por ejemplo en caso de que anidáramos dentro de un ciclo «Mientras» a un ciclo «Para», el ciclo «Para» se terminaría primero.



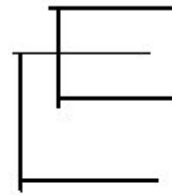
a) Independientes



b) Anidados

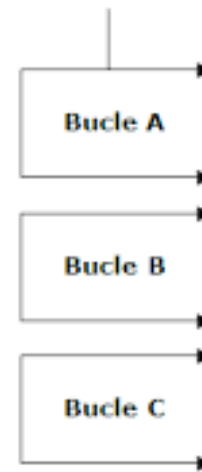


c) Cruzados

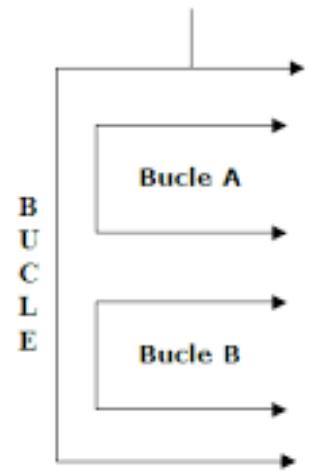


No es correcto su diseño

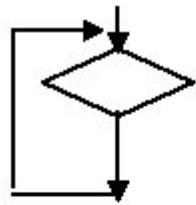
BUCLES SENCILLOS
INICIO



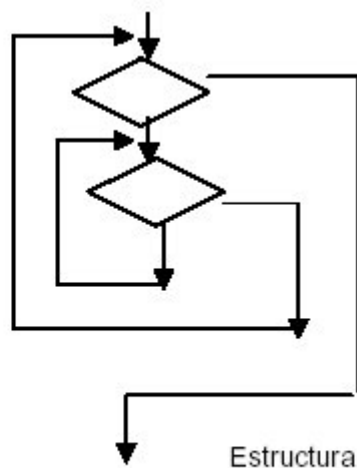
BUCLES ANIDADOS
INICIO



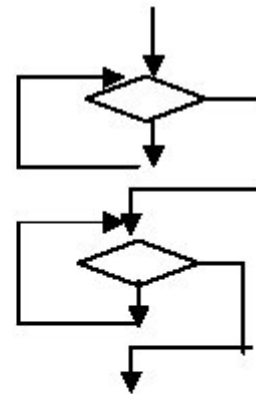
Diseño de las estructuras en flujograma:



Estructura repetitiva simple



Estructura repetitiva Anidada



Estructura repetitiva Compuesta múltiple

EUPs que imprima la tabla de multiplicar de un numero proporcionado del el usuario



Entero tabla

Escribir "numero de la tabla a imprimir"

Leer tabla

Para $i=1$, $i \leq 10$ $i=i+1$ hacer

- Escribir tabla, "X", i, "=" tabla*i

Fin_para

Factorial de un numero

¿Qué es la función Factorial?

La función factorial es una fórmula matemática representada por el signo de exclamación "!". En la fórmula Factorial se deben multiplicar todos los números enteros y positivos que hay entre el número que aparece en la fórmula y el número 1.

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

$$1! = 1$$

$$2! = 2 \times 1 = 2$$

$$3! = 3 \times 2 \times 1 = 6$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
para(k=1;k<=i;k++)  
  {  
    fact=fact*i  
  }
```

Algoritmos

Ciclos

$$1!+2!+3!+4!+\dots+N!$$

Algoritmo SumaFactorial

fact, suma=0,i,k, N

Leer (N)

```
para(i=1;i<=N;i++)
```

```
{
```

```
    para(k=1;k<=i;k++)
```

```
    {
```

```
        fact=fact*i
```

```
    }
```

```
    suma=suma+fact
```

```
    fact=1
```

```
}
```

```
Imprimir (“El resultado es”, suma)
```

Algoritmos

Ciclos

$$1^1 + (2^1 + 2^2) + (3^1 + 3^2 + 3^3) + \dots + (N^1 + N^2 + \dots + N^N)$$

i,j,k,N,suma=0,suma2

Leer (N)

for (i=1;i<=N;i++)

{

for (j=1, suma2=0 ;j<=i;j++)

{

for (k=1,pot=1;k<=j;k++)

{

pot=pot*i

}

suma2=suma2+pot

}

suma=suma+suma2

}

Imprimir (“El resultado es”, suma)

Algoritmos

Tipos de Datos

- int (Enteros)
 - float (Flotantes o decimales)
 - char (caracteres)
 - void (vacío)
 - bool (booleano o lógico)
-
- Estos son los tipos básicos de datos. Las variables utilizadas en un algoritmo y con mayor razón en un programa un nombre (etiqueta, label) y un tipo de dato específico.

Algoritmos

Es importante destacar que las operaciones entre tipos de datos, retornan valores del mismo tipo.

Por ejemplo

- `int i=4, j=2`
- `i/j` es igual a 2

Pero

- `int i=5, j=2`
- `i/j` es igual a 2

Algoritmos

Ejemplo

- float $i=5.0$, $j=2.0$
- i/j es igual a 2.5

Caracteres

- char $a='h'$, $b='o'$, $c='l'$, $d='a'$
- Imprimir (a,b,c,d)
 - Su salida es hola
- Nos podemos fijar que al definir un carácter, este sólo puede almacenar un símbolo. Al igual que una variable entera o real sólo puede almacenar un valor.

Vectores

Los vectores son una colección de variables del mismo tipo, que tienen un nombre común.

También son llamados arreglos. Los cuales son definidos por un nombre, dimensión y tipo de dato.

Para acceder a un elemento particular de un arreglo se usa su nombre y su ubicación en el conjunto (denominado índice).

Vectores – arreglos unidimensionales

Para declarar un arreglo:

- tipo nombre[dimensión]

Ejemplo:

- `int notas[5], a[10]`
- `float promedios[40]`

Vectores – arreglos unidimensionales

int notas[5]

Notas (el contenido)

Índices del arreglo

60	59	15	23	77
0	1	2	3	4

En realidad podemos visualizar el arreglo como:

60	59	15	23	77
----	----	----	----	----

Vectores – arreglos unidimensionales

Para acceder a un elemento del arreglo utilizamos el siguiente formato:

- Nombre_arreglo[indice]
- Por ejemplo:

60	59	15	23	77
----	----	----	----	----

- Imprimir (notas[3])
 - Debería desplegar: 23
- terceraNota=notas[3]
- Imprimir (terceraNota)
 - Debería desplegar : 23

Vectores - Cadenas

Para representar una cadena (conjunto de caracteres) se utiliza un arreglo o vector de caracteres

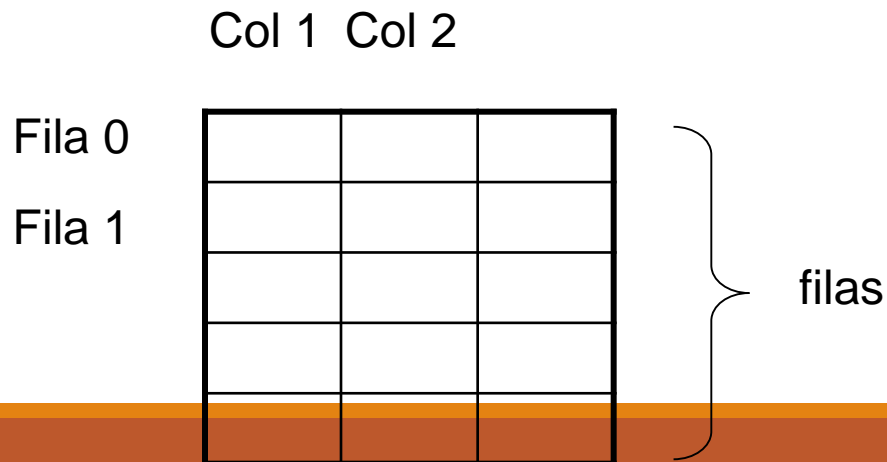
- `char nombre[30]`
- `char direccion[80]`

<code>nombre[0]='h'</code>		<code>char nombre="hola"</code>
<code>nombre[1]='o'</code>	→	
<code>nombre[2]='l'</code>		<code>char nombre={'h','o','l','a'}</code>
<code>nombre[3]='a'</code>		

Arreglos multidimensionales

Son arreglos de más de una dimensión

- Se definen como:
- **Tipo nombre[fila][columna]**
- Por ejemplo:
- `int matriz[5][3]`



Ejemplo

10	20	30
40	50	60
70	80	90
100	110	120
130	140	150

Para acceder al elemento que contiene 150

`matriz[4][2]`

Ejercicios: Qué realiza el siguiente algoritmo sobre un vector de dimensión 5 cuyos elementos son ingresados por teclado

```
int i,s,a[5],m;
float p;
for ( i=0; i<5 ; i=i+1)
{
    Imprimir ("Ingrese un elemento")
    leer(a[i])
}
s=0; m=1;
for ( i=0; i<5 ; i++)
{
    s = s + a[i];
    m = m * a[i];
}
p=m/s
Imprimir("la suma , multiplicación y resultado?? es",s,m,p)
```

Tarea1

Programa para imprimir un rectángulo de asteriscos (*) leyendo el tamaño del lado y la altura con un máximo de 20.

Ejemplo:

Si se tecllea 10 8, imprimirá

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

```
Algoritmo Rect
variables i,j, Lado, altura ENTERO
INICIO
REPETIR
ESCRIBIR "DAME EL VALOR DE LADO"
LEER Lado
ESCRIBIR "DAME EL VALOR DE LA ALTURA"
LEER Altura
MIENTRAS Altura > 20 ENTONCES
  DESDE (i=0; i < Altura; i=i+1)
    DESDE (j=0; j < Lado; j=j+1)
      ESCRIBIR("*")
    FIN_DESDE
  ESCRIBIR("\n")
  FIN_DESDE
FIN
```

Tarea2

Se desea un programa que funcione como una calculadora._____

El programa leerá Op1, Op2 y tipoOperacion

Donde TipoOperacion puede ser: +, -, *, / y ^ (potencia)

Ejem.: 5.7+4.2, 6.7/2.5,

El programa terminará los 2 operadores sean CERO

El programa puede hacerse con un esquema de ciclo con centinela.

Algoritmo Rect

variables i,j, Lado, altura ENTERO

INICIO

REPETIR

ESCRIBIR "DAME EL VALOR DE LADO"

LEER Lado

ESCRIBIR "DAME EL VALOR DE LA ALTURA"

LEER Altura

MIENTRAS Altura > 20 ENTONCES

DESDE (i=0; i < Altura; i=i+1)

DESDE (j=0; j < Lado; j=j+1)

ESCRIBIR("*")

FIN_DESDE

ESCRIBIR(\n)

FIN_DESDE

FIN

Tarea3

EUPs que imprima la tabla de multiplicar de 2 al 10

Algoritmo Tablas

variables i,j, ENTERO

INICIO

DESDE (i=2; i < 10; i=i+1)

DESDE (j=1; j < 10; j=j+1)

ESCRIBIR(i," X ",j, " = ", i*j)

FIN_DESDE

ESCRIBIR(\n)

FIN_DESDE

FIN