


ESTRUCTURAS DE REPETICIÓN

Ciclos Anidados:

Dependiendo del problema se puede tener un ciclo dentro de otro, cuando esto ocurra tenga encuenta lo siguiente, el ciclo mas interno debe finalizar primero, para que el mas externo pueda volver a incrementarse.

	ciclo externo	para (i=1,i<=2; i=i+1)	i	j	(i<=2)	(j<=3)	suma
			1	1	(1<=2) C	(1<=3) C	0 +1=1
	ciclo interno	para (j=1; j<=3, j=j+1)	1	2		(2<=3) C	1+2=3
			1	3		(3<=3) C	3+3=6
		suma= suma + j;		4		(4<=3) F	
		imprimir (suma)	2	1	(2<=2) C	(1<=3) C	6+1=7
			2	2		(2<=2) C	7+2=9
			2	3		(3<=3) C	9+3=12
				4		(4<=3) F	
			3		(3<=2) F		

Cualquiera de los ciclos anteriores pueden ser anidados, eso si no olvide como trabaja este proceso, veamos un ejemplo:

3. Ciclos anidados

Definición del Problema: Elabore un algoritmo que imprima las tablas de multiplicar impares(1..5), con 10 elementos cada una. Cada vez que se de una nueva tabla imprima un titulo que indique la tabla que se va a iniciar. Y cada elemento debe tener el siguiente formato:

1*1=1
1*2=2

Análisis y Diseño

Entrada	
Proceso	Generar tablas de multiplicar impares de 1 a 5 (mult) Calcular cada tabla de 1 a 10 (multp) (prod) = mul t* multp
Salida	mult, multp, prod

Algoritmo Multiplicacion {

// Bloque de declarativas

entero mult, multp, prod;

//Bloque del Instrucciones

prod = 0;

para(mult =1; mult<=5; mult=mult+2) ←ciclo externo

{

imprimir("Tabla del ", mult);

para (multp=1; multp<=10; multp = multp+1){

prod = mult * multp;

imprimir(mult, " * ", multp, "=", prod);

}

}

}

Ciclo
interno

3. Ciclos anidados

Definición del Problema: Elabore un algoritmo que permita realizar divisiones de números enteros, recuerde validar que el denominador no puede ser negativo, antes de realizar el calculo imprima el numerador, denominador y respuesta .

Análisis y Diseño

Entrada	nume, deno, resp
Proceso	Repetir mientras el usuario lo desee(res) Validar denominador no sea cero (multp) Calcular (fracción) = nume/deno
Salida	nume, deno, prod

```
Algoritmo Fraccion {  
  // Bloque de declarativas  
  entero nume, deno, prod;  
  caracter res;  
  //Bloque del Instrucciones  
  fraccion = 0;  
  res = 's';  
  Mientras(res == 's' ) ← ciclo externo  
  {  
    imprimir( "Ingrese numerador y denominador");  
    leer( nume, deno );  
    mientras( deno == 0 ) { ← ciclo interno  
      imprimir( "denominador, no puede ser cero");  
      leer( deno); }  
    fraccion = nume/ deno;  
    imprimir(nume, "/", deno, "=", fraccion);  
    imprimir( "Desea probar otros datos? s, n");  
    leer(res); } }  
}
```