

## Conceptos preliminares para la programación

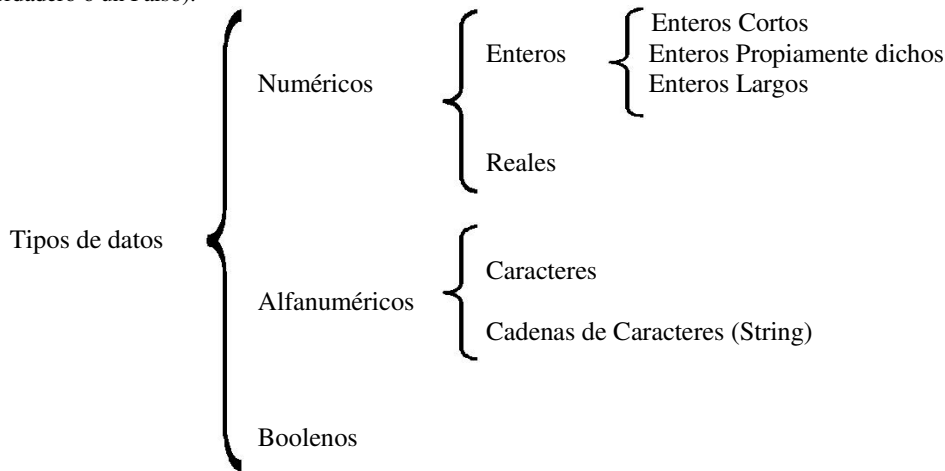
**Algoritmo:** Conjunto de pasos lógicos, secuenciales y finitos para dar solución a un problema general.

**Dato:** Es una característica de un objeto o entidad, que adquiere un valor. Por ejemplo, si se piensa en los datos asociados a un estudiante (la entidad es el estudiante), podemos identificar datos como su nombre, edad, sexo, estado civil, fecha de nacimiento, y muchos otros más datos, que pueden relacionarse con el estudiante.

**Constante:** Un dato se dice que es constante si su valor nunca cambia dentro de un algoritmo.

**Variable:** Una variable es un dato cuyo valor cambia por lo menos una vez dentro de un algoritmo. Toda variable debe ser identificada mediante un nombre único dentro de un algoritmo. Los nombres de las variables son palabras descriptivas del dato que contienen, sin espacios ni signos de puntuación, usando solo letras del alfabeto inglés y números, comenzado siempre en letra.

**Tipos de datos:** Los datos se clasifican según el tipo de valor que contenga así: Numéricos (si su contenido es un número) Alfanuméricos (si su contenido es un carácter o una cadena de caracteres) y Booleanos (si su contenido es un valor de verdad, es decir un verdadero o un Falso).



**Expresión:** Es una combinación apropiada de operandos, operadores, signos de agrupación y funciones. Los operandos son datos constantes o variables, los signos de agrupación son los paréntesis y las funciones son las predefinidas por el lenguaje o construidas por el programador.

Nota: en este curso sólo está permitido el uso de los operadores aquí definidos.

		Suma +			
		Resta -			
		División real /			
	Aritméticos	Multiplicación *			
		División entera div			
		Residuo de división entera mod			
	Lógicos	{ And	X	Y	X and Y
		Or	V	V	V
		not	V	F	F
			F	V	V
			F	F	F
			F	F	V
	Relacionales	{ Mayor >			
		Menor <			
		Mayor o igual ≥			
		Menor o Igual ≤			
		Igual ==			
		Diferente ≠			

Operadores

**Evaluar una expresión:** Es el proceso mediante el cual se realizan las operaciones indicadas en una expresión, con los valores que en ese instante tengan las variables, respetando el orden que plantean los signos de agrupación y obteniendo un único valor que puede ser un número, un carácter o cadena de caracteres, o un valor de verdad (un falso o un verdadero)

**Tipos de expresión:** Una expresión puede ser numérica, alfanumérica o booleana. Es numérica si después de evaluar (calcularla expresión) su resultado es un número; es alfanumérica si después de evaluar la expresión el resultado es un carácter o cadena de carácter y es booleana si después de evaluar la expresión su resultado es un valor de verdad (un verdad o un falso)

Ejemplos: si a, b y c son datos numéricos

$(a+b)*c$  es una expresión numérica

$((a+b)*c) > 5$  es una expresión booleana

$((a>b) \text{ and } (c>a+b)) \text{ or } (c > 3)$  es una expresión booleana

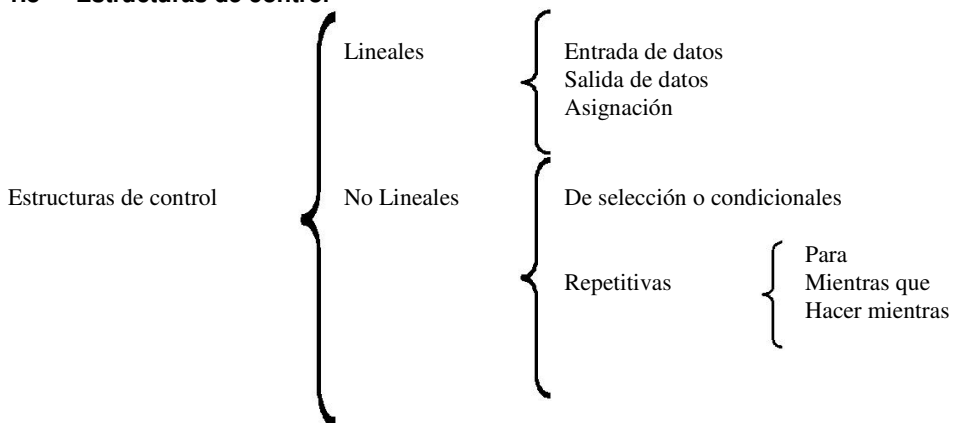
$(a \text{ div } 4) = 0$  es una expresión booleana

$(a \text{ mod } 5) > (b \text{ div } 4)$  es una expresión booleana

$((a \text{ mod } 5) > 4)*c$  no es una expresión válida

$((a \text{ mod } 5) - 4)*c$  es una expresión numérica

### 1.8 Estructuras de control



## 2 Estructuras de control lineales (Entrada, Asignación y Salida)

Estructuras de control			
Nombre	Diagrama de flujo	Seudocódigo	En C
Entrada o captura de datos		Leer b	<code>scanf ("%d", &amp;b);</code>
Salida o impresión de datos		Imprimir a	<code>printf ("cadena de formateo ", a);</code>
Asignación		<code>a = b + c</code>	<code>a = b + c;</code>

En una asignación hay que tener en cuenta:

Variable cuyo valor va a cambiar

Símbolo de asignación

$$a = b + c$$

En una asignación, el tipo de dato de la variable de la izquierda debe ser igual al tipo de expresión de la derecha

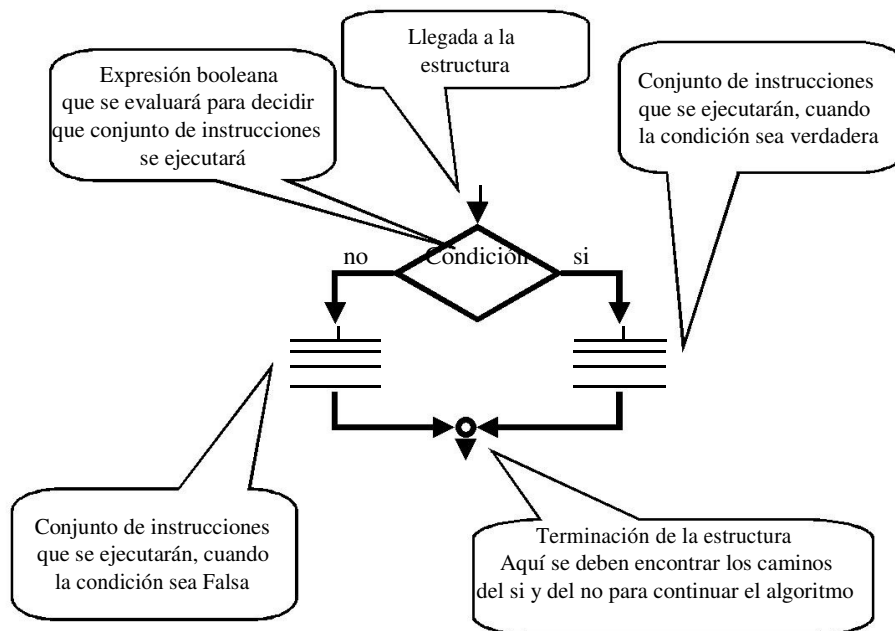
En una asignación, primero se calcula el valor de la expresión y luego el resultado es almacenado en la variable de la izquierda

Expresión que va a ser calculada, y cuyo resultado será almacenado en la variable de la izquierda

### 3 Estructuras de selección o Condicionales

	<p>Si condición entonces          ....Instrucciones para          ....cuando la condición          ....sea verdadera</p> <p>Sino          ....Instrucciones para          ....cuando la condición          ....sea falsa</p> <p>Fin si</p>	<pre>if ( condición ) {     ....Instrucciones en c para     ....cuando la condición     ....sea verdadera } else {     ....Instrucciones en c para     ....cuando la condición     ....sea falsa };</pre>
--	--	---

En una estructura de selección hay que tener en cuenta:



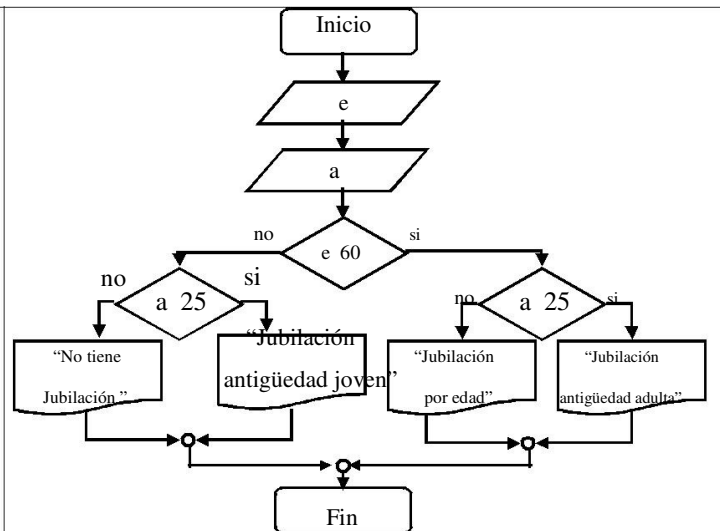
- -El Seguro Social requiere clasificar a las personas que se jubilarán en el año de 2005. para este programa el seguro social plantea las siguientes reglas:

Existen tres tipos de jubilaciones: por edad, por antigüedad joven y por antigüedad adulta. Las personas adscritas a la jubilación por edad deben tener 60 años o más y una antigüedad en su empleo de menos de 25 años. Las personas adscritas a la jubilación por antigüedad joven deben tener menos de 60 años y una antigüedad en su empleo de 25 años o más. Las personas adscritas a la jubilación por antigüedad adulta deben tener 60 años o más y una antigüedad en su empleo de 25 años o más. Determinar en que tipo de jubilación, quedara adscrita una persona conociendo su edad y tiempo de trabajo.

Solución	
Análisis: Entrada: e: edad del empleado a: antigüedad del empleado en la empresa Salidas: Mensaje con el tipo de jubilación	
Diseño	
Seudocódigo	Diagrama de Flujo
Inicio imprimir "digite la edad del empleado" leer e imprimir "digite la antigüedad del empleado" leer a si e >= 60 entonces si a >=25 entonces imprimir "jubilación por antigüedad adulta" Sino imprimir "jubilación por edad" Finsi Sino si a >= 25 entonces imprimir "jubilación por antigüedad joven" Sino	

imprimir "no tiene jubilación"

fin si  
fin si  
fin



### Codificación

```
#include <stdio.h>
int main() {
    int e,a;
    printf("digite la edad del empleado \n");
    scanf("%d", &e);
    printf("digite la antigüedad del empleado \n");
    scanf("%d",&a);
    if (e >= 60) {
        if (a >= 25){
            printf("jubilación por antigüedad adulta \n");
        } else {
            printf("jubilación por edad \n");
        };
    } else {
        if (a >= 25) {
            printf("jubilación por antigüedad joven \n");
        } else {
            printf("no tiene jubilación \n");
        };
    };
    return (0);
}
```

- Capturar por teclado tres números cualquiera (A, B y C) que corresponden a las aristas de un triángulo; el programa debe determinar que tipo de triángulo, teniendo en cuenta la siguiente clasificación:

Si los tres lados son iguales es equilátero

Si solo dos lados son iguales isósceles

Si todos son diferentes escalenos

Solución

Análisis:

Entrada: A,B,C: medidas de las tres aristas de un triángulo

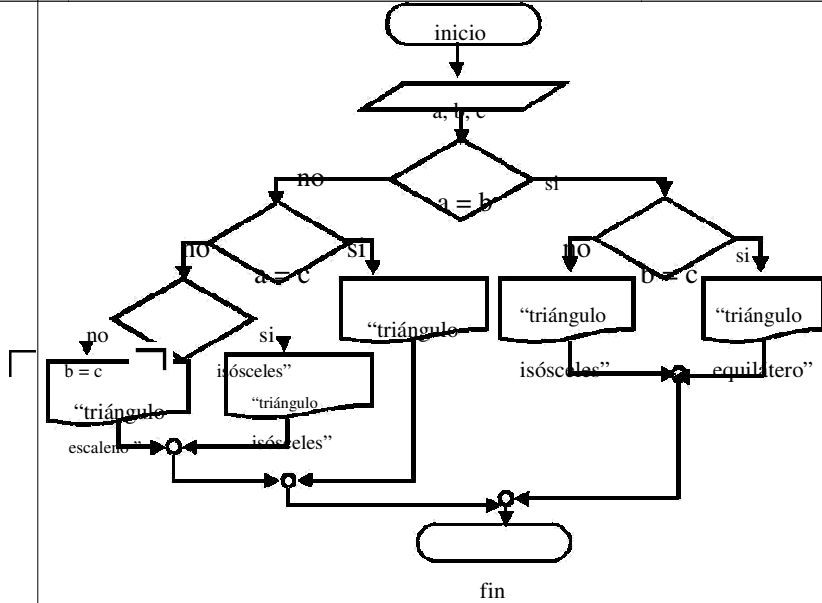
Salidas: Mensaje con el tipo de triángulo

Diseño

Seudocódigo

```
inicio
imprimir "digite el valor de la arista1"
leer a
imprimir "digite el valor de la arista2"
leer b
imprimir "digite el valor de la arista3"
leer c
si a = b entonces
  si b = c entonces
    imprimir "triángulo equilátero"
  sino
    imprimir "triángulo isósceles"
  fin si
sino
  si a = c entonces
    imprimir "triángulo isósceles"
  sino
    si b = c entonces
      imprimir "triángulo isósceles"
    sino
      imprimir "triángulo escaleno"
  fin si
fin si
fin
```

Diagrama de flujo



fin si

fin

Codificación en C

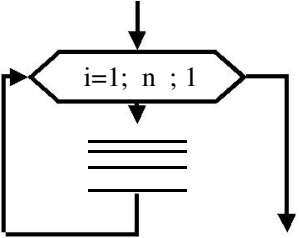


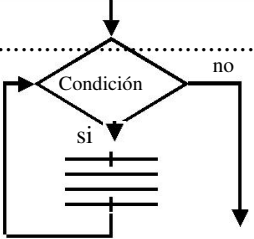
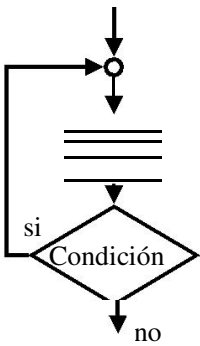
```

#include <stdio.h>
int main() {
    int a,b,c;
    printf("digite el valor de la arista1\n");
    scanf("%d", &a);
    printf("digite el valor de la arista2\n");
    scanf("%d", &b);
    printf("digite el valor de la arista3\n");
    scanf("%d", &c);
    if (a == b) {
        if (b == c) {
            printf("triángulo equilátero \n");
        } else{
            printf("triángulo isósceles \n");
        };
    } else {
        if (a == c) {
            printf("triángulo isósceles \n");
        } else {
            if (b == c) {
                printf("triángulo isósceles \n");
            } else {
                printf("triángulo escaleno \n");
            };
        };
    };
    return (0);
} // aquí termina el main

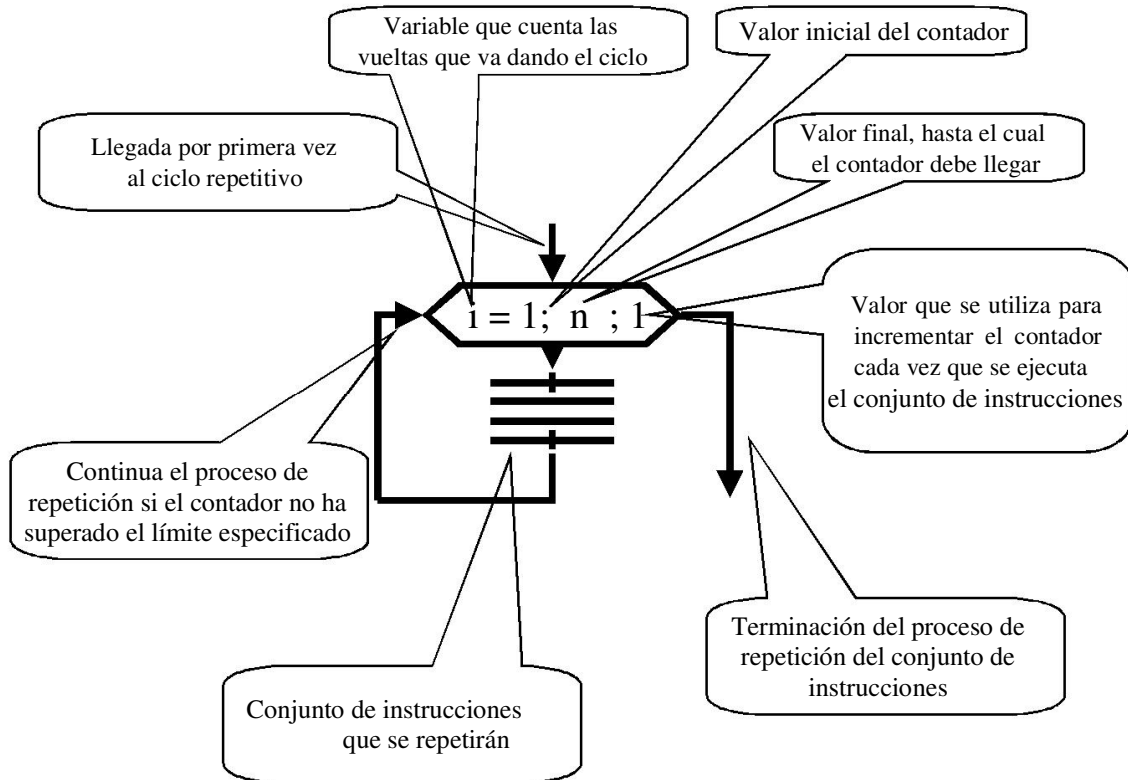
```

#### 4 Estructuras repetitivas

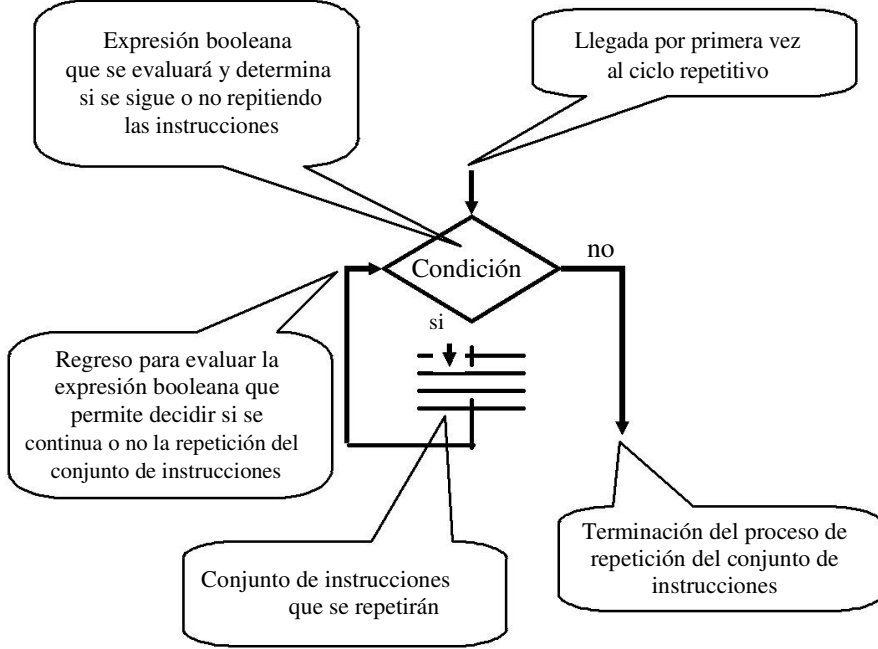
<p>Estructura repetitiva Para</p>		<p>Para i=1 hasta n inc 1 haga ..... Instrucciones a repetir fin para</p>	<pre>for ( i =1 ; i&lt;=n ; i=i+1 ){ ..... Instrucciones a repetir };</pre>
<p>Estructura repetitiva Mientras que</p>		<p>MQ condición haga</p>	<pre>while ( condición ) {</pre>

Instrucciones a repetir fin MQ			Instrucciones a repetir };
Estructura repetitiva Hacer Mientras		Hacer .... Instrucciones a repetir MQ condición	Do { .... Instrucciones a repetir } while ( condición );

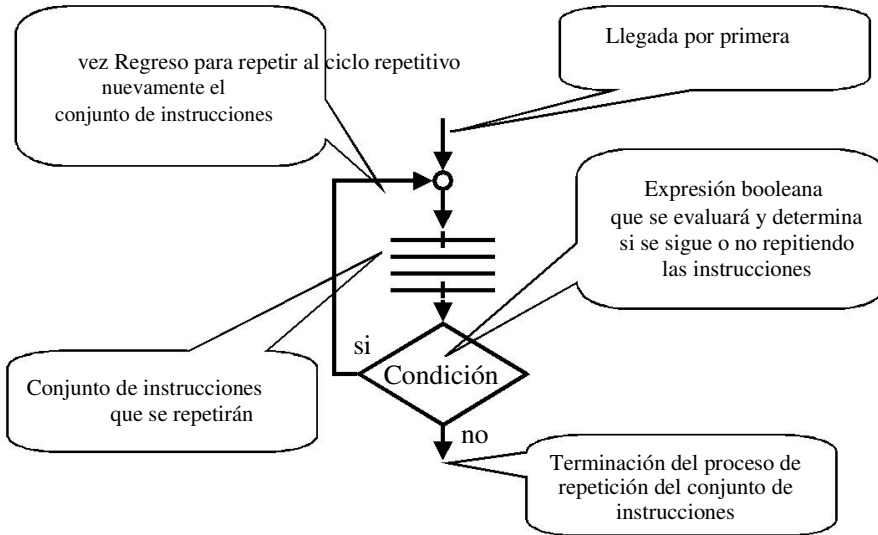
En una estructura repetitiva para hay que tener en cuenta:



En una estructura repetitiva Mientras Que (MQ), hay que tener en cuenta:



En una Estructura repetitiva Hacer Mientras, hay que tener en cuenta:



- Dado un número calcular e imprimir si es primo

Solución	
<b>Análisis</b> Entradas: n: un entero positivo del cual se desea saber si es o no primo Salidas: Un mensaje que diga si el número dado es primo o no	
Diseño en pseudocódigo	Codificación en lenguaje C
Inicio leer( n ) e=0 para i =1 hasta n hacer si ( ( n mod i ) == 0 ) e = e+1 fini finpara si ( e <= 2 ) entonces imprimir( "el número es primo" ) Sino imprimir( "el número no es primo" ) Finsi Fin	<pre>#include&lt;stdio.h&gt; int main(){   long int n,e, i;   printf("digite un número ");   scanf("%ld", &amp;n);   e=0;   for( i =1; i &lt;=n; i = i+1){     if ( ( n % i ) ==0 ){       e=e+1;     }else{     };   };   if ( e &lt;=2){     printf("El número %ld es primo", n);   }else{     printf("El número %ld No es primo", n);   };   system("pause");   return (0); }</pre>

## Ejercicios

**IF** //Hacer un programa que muestre en pantalla si un numero ingresado por teclado es positivo..o negativo.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

main ()
{
  int x;
  printf("Ingrese un numero\n");
  scanf("%d",&x);

  if (x>0) {
    printf("El numero %d es positivo\n",x);
  }
  //if (x<0){
  Else                   IF-ELSE

  printf("El numero %d es negativo\n",x);//}
  getch();
  system("PAUSE");}
```

## **WHILE**

```
//Hacer programa que sume los números solicitados por teclado
//mientras no sean negativos

#include <stdio.h>
#include <stdlib.h>
```

```
main ( )
{
int Suma, Num;
Suma = 0;
printf( "Ingresa un número");
scanf("%d", &Num);
while(Num >= 0)
{
Suma = Suma + Num;
printf( "\nIngresa otro número");
scanf("%d", &Num);
}
printf("\n Suma = %d\n", Suma);
system("PAUSE");
}
```

### **FOR**

```
//Mostrar los números del 1 al 100.-----DEL 100 AL 1
#include <stdio.h>
#include <stdlib.h>

main(){ int x;
        for (x=1;x<=100;x++)-----//solo cambiar for (x=100;x>=1;x++){
        printf("%d\n", x);}
system("PAUSE");
return 0;}
```