

Construcciones algorítmicas

En C existen estas construcciones para implementar algoritmos. ***Sentencias simples y sentencias compuestas.***

Sentencias.- Es un fragmento de código.

Una sentencia simple es una expresión terminada en punto y coma.

sentencia_simple;

Una sentencia compuesta es una serie de sentencias simples entre llaves.

/* sentencia compuesta: varias sentencias entre llaves*/

{

Sentencia1

Sentencia2

...

SentenciaN

}

Construcciones Condicionales-Selección

Sentencia if

La construcción if sirve para ejecutar código, sólo si una condición es cierta:

if (condición)

sentencia

En las expresiones comparativas se necesitan los operadores lógicos.

Operadores booleanos

Operador	
$A == B$	si A es igual a B
$A != B$	si A es distinto de B
$A > B$	si A es mayor que B
$A < B$	si A es menor que B
$A >= B$	si A es mayor o igual que B
$A <= B$	si A es menor o igual que B

Ejem5.

// Leer un número X e imprimir si es cero.

```
#include <stdio.h>  
int main( )  
{ int x;  
printf("\n\tIngrese un valor numérico para x:");  
scanf("%d", &x);  
if ( x == 0 )  
printf ("\n El valor de tu variable es cero\n");  
return 0;  
}
```

```
// Nombre del archivo: ejem5.c  
// Compilado: gcc ejem5.c -o ejem5  
//Ejecutado: ./ejem5
```

Sentencia if –compuesta-

if (condición)

{

sentencia1

sentencia2

}

Ejem6.

*/*Leer un número X e imprimir si es cero. */*

#include <stdio.h>

int main()

{

int x;

printf("\n\tIngrese un valor numérico para x:");

scanf("%d", &x);

if (x == 0){

printf ("\n El valor de tu variable es cero\n");

printf ("\n Segunda instrucción de if verdadero\n");

```
}// cierre de if verdadero, uso de sentencia compuesta  
return 0;  
}//cierre de main  
// Nombre del archivo: ejem6.c  
// Compilado: gcc ejem6.c -o ejem6  
//Ejecutado: ./ejem6
```

Construcción if-else

Con la construcción else se pueden definir acciones para cuando la condición del if sea falsa.

La sintaxis es

```
if (condición)  
    sentencia1  
  
else  
    sentencia2
```

Ejem7.

*/*Leer un número Z e imprimir si es negativo o positivo. */*

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

```
int main( )
```

```
{
```

```
    int x;
```

```
    printf("\n\tIngrese un valor numérico para x:");
```

```
    scanf("%d", &x);
```

```
    if ( x == 0 )
```

```
        printf ("\n El valor de tu variable es cero\n");
```

```
    else
```

```
if(x>0)  
    printf ("\n El valor de tu variable es positivo\n");  
else  
    printf ("\n El valor de tu variable es negativo\n");  
return 0;  
// cierre de main  
// Nombre del archivo: ejem7.c  
// Compilado: gcc ejem7.c -o ejem7  
//Ejecutado: ./ejem7
```

Construcción if-else –compuesta-

Ejem8.

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

```
int main( )
```

```
{ int x;
```

```
printf("\n\tIngrese un valor numérico para x:");
```

```
scanf("%d", &x);
```

```
if ( x == 0 ){
```

```
printf ("\n El valor de tu variable es cero\n");
```

```
printf ("\n %d\n",x); }
```

else

if(x>0){

*printf ("\n El valor de tu variable es **positivo**\n");*

printf ("\n %d \n",x);}

else{

*printf ("\n El valor de tu variable es **negativo**\n");*

printf ("\n %d \n",x);}

return 0;

// Nombre del archivo: ejem8.c

// Compilado: gcc ejem8.c -o ejem8

//Ejecutado: ./ejem8

Para elaborar condiciones complejas, existen estos operadores:

Expresión	Resultado
E1 && E2	Cierta si E1 y E2 son ciertas (AND)
E1 E2	Cierta si E1 o E2 son ciertas (OR)

Construcción switch

Se utiliza para ejecutar acciones diferentes según el valor de una expresión.

Ejemplo de sintaxis:

```
switch (expresión)
{
    case valor1:
        ... sentenciasA...
    break;
```

case valor2:

... sentenciasB ...

break;

.

.

.

default:

... sentenciasN ...

break;

}

Las sentenciasA se ejecutarán si la expresión adquiere el valor1.

Las sentenciasB se ejecutarán si adquiere el valor2.

Cualquier otro valor de la expresión conduce a la ejecución de las sentenciasN; eso viene indicado por la palabra reservada **default**.

Ejem9.

`/*menuA`: Escribir un código fuente que muestre el siguiente menú al usuario, indique al usuario que deberá elegir solo una opción a realizar.

Opción 1 -- Suma

Opción 2 -- Resta

Opción 3 -- Multiplicación

Opción 4 -- División

Opción default –Mostrar “verifique su selección”`*/`

```
#include <stdio.h>  
int main(){  
    int op;  
    float a,b,res;  
    printf("\n\t FAVOR DE SELECCIONAR UNA OPCION\n");  
    printf("\n\t OPCION DE SUMA 1\n");  
    printf("\n\t OPCION DE RESTA 2\n");  
    printf("\n\t OPCION DE MULTIPLICACION3\n");  
    printf("\n\t OPCION DE DIVISION 4\n\n\t\t\t");  
    scanf("%d",&op);  
    switch(op) {  
        case 1:
```

```
printf("\n\t TECLEAR EL VALOR DE A: \t");  
scanf("%f",&a);  
printf("\n\t TECLEAR EL VALOR DE B: \t");  
scanf("%f",&b);  
res=a+b;  
printf("\n\t La suma es : %f\n", res);  
break;  
case 2:  
printf("\n\t TECLEAR EL VALOR DE A: \t");  
scanf("%f",&a);  
printf("\n\t TECLEAR EL VALOR DE B : \t");  
scanf("%f",&b);
```

```
res=a-b;
printf("\n\t La resta es : %f\n", res);
break;
case 3:
printf("\n\t TECLEAR EL VALOR DE A: \t");
scanf("%f",&a);
printf("\n\t TECLEAR EL VALOR DE B: \t");
scanf("%f",&b);
res=a*b;
printf("\n\t La multiplicación es : %f\n", res);
break;
case 4:
```

```
printf("\n\t TECLEAR EL VALOR DE A: \t");  
scanf("%f",&a);  
printf("\n\t TECLEAR EL VALOR DE B: \t");  
scanf("%f",&b);  
if(b!=0) {  
    res=a/b;  
    printf("\n\t La división: %f\n", res);  
    //sentencia doble  
else  
    printf("\n\t ERROR!!!\n");  
break;
```

default:

printf("\n\t OPCION NO VALIDA\n");

break;

//cierre de switch

return 0;

//cierre de main

// Nombre del archivo: menuA.c

// Compilado: gcc menuA.c -o menuA

//Ejecutado: ./menuA