Estructuras de Repetición

Ciclo while

Para ejecutar el mismo código varias veces, se puede utilizar:

```
while ( condición ){
    sentencias
}
```

Las sentencias se ejecutan una y otra vez mientras la condición sea cierta.

Ejem10.

```
// Generar  e imprimir los números del  1, 2, 3, hasta el 10.
#include <stdio.h>
int main()
{
   int x=1;
   while ( x <= 10 )
   {
        printf("%d\n",x);
        x=x+1;
   }
return 0;}// Nombre del archivo: ejem10.c
```

Ejem11.

```
//Generar e imprimir los primeros 20 números pares.
#include <stdio.h>
int main()
{
   int par=2;
   while ( par <= 40 )
   {
        printf("%d\n",par);
        par=par+2;
   }
return 0;}// Nombre del archivo: ejem11.c
```

Ejem12.

```
//Generar e imprimir los primeros 30 números impares.
#include <stdio.h>
int main()
{
   int impar=1;
   while ( impar <= 59 )
   {
        printf("%d\n",impar);
        impar=impar+2;
   }
return 0;}// Nombre del archivo: ejem12.c
```

Ejem13.

```
/*Hallar e imprimir los cuadrados de la numeración del 100, 99,
   98 hasta el 1.
   100, 10000
   99, 9801 */

#include <stdio.h>
int main()
{
   int x=100,c;
   while ( x>= 1)
   {
```

```
        c=x*x;
        printf("%d, %d\n",x,c);
        x=x-1;
    }
return 0;}// Nombre del archivo:ejem13.c
```

Ciclo for

También existe el ciclo “for”, según esta sintaxis:

```
for ( expresión_inicial; condición; expresión_de_paso )
{
        sentencias ejecutables
}
```

La expresión_inicial se ejecuta antes de entrar en el ciclo.

Si la condición es cierta, se ejecutan las sentencias y después expresión_de_paso.

Luego se vuelve a evaluar la condición, y así se ejecuta la sentencia una y otra vez hasta que la condición sea falsa.

Ejem14.

```
// Generar  e imprimir los números del  1, 2, 3, hasta el 10.
#include <stdio.h>
int main()
{
    int x;
  for(x=1; x<= 10; x=x+1 )
    {
        printf("%d\n",x);
    }
return 0;}// Nombre del archivo: ejem14.c
```

Ejem15.

```
//Generar e imprimir los primeros 20 números pares.
#include <stdio.h>
int main()
{
    int par;
    for(par=2; par <= 40; par=par+2)
    {
        printf("%d\n",par);
    }
return 0;}// Nombre del archivo: ejem15.c
```

Ejem16.

```
//Generar e imprimir los primeros 30 números impares.
#include <stdio.h>
int main()
{
    int impar;
    for(impar=1; impar <= 59; impar=impar+2 )
    {
        printf("%d\n",impar);
    }
return 0;}// Nombre del archivo: ejem16.c
```

Ejem17.

/*Hallar e imprimir los cuadrados de la numeración del 100, 99, 98 hasta el 1.

100, 10000

99, 9801 */

```
#include <stdio.h>
int main()
{
    int x,c;
```

```
    for(x=100; x>= 1; x=x-1)
    {
        c=x*x;
        printf("%d, %d\n",x,c);
    }
return 0;}// Nombre del archivo: ejem17.c
```

Operadores avanzados

Los operadores de incremento, decremento y asignación compuesta permiten modificar el contenido de una variable de forma eficiente y abreviada.

Operadores	Significado
A++, ++A	Incrementa en 1 el valor de A (A=A+1)
A--, --A	Disminuye en 1 el valor de A (A=A-1)

Operadores	Significado
A+=x	A=A+x
A-=x	A=A-x
A*=x	A=A*x
A/=x	A=A/x

Operadores “pre” y “post” y valor devuelto

Si el operador ++ o -- se coloca a la izquierda, se llama preincremento o predecremento, respectivamente. Si se coloca a la derecha, se llama postincremento o postdecremento.

Operaciones “pre”: El valor nuevo de la variable afectada

Operaciones “post”: el valor anterior de la variable afectada.

Uso de operadores avanzados, cambiar los ejemplos anteriores usando los operadores ++ y --.

Ciclo do-while

Sintaxis:

```
do {
    sentencias ejecutables.
} while ( condición );
```

Las sentencias se ejecutan al menos la primera vez; luego, mientras la condición sea cierta, se iteran las sentencias ejecutables.

El ciclo do-while lo podemos usar para los menús recursivos.

Ejem18.

```
/*Menú con los últimos 4 ejercicios uso de for y operadores avanzados*/

#include <stdio.h>

int main(){

char menu, R;

int x, par, impar,c;
```

```
do{
printf("\n\t FAVOR DE SELECCIONAR UNA OPCIÓN\n");
printf("\n\t -a- ejem 14\n");
printf("\n\t -b- ejem 15\n");
printf("\n\t -c- ejem 16\n");
printf("\n\t -d- ejem 17\n");
scanf("%s",&menu);
switch(menu) {
  case 'a':
      for(x=1; x<= 10; x++ )
```

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


    }
break;
case 'b':
    for(par=2; par <= 40; par+=2)
    {
        printf("%d\n",par);
    }
```

```
break;
case 'c':
    for(impar=1; impar <= 59; impar+=2 )
    {
        printf("%d\n",impar);
    }
break;
case 'd':
     for(x=100; x>= 1; x--)
    {
```

```
        c=x*x;
        printf("%d, %d\n",x,c);
    }
 break;

 default:
    printf("\n\t OPCION NO VALIDA\n");
break;
}//cierre de switch
```

```
 printf("\n\t DESEA VER EL MENU NUEVAMENTE Escriba s o
S\n");

 scanf("%s",&R);

 }while(R=='S' || R=='s' );//cierre de do-while

return 0;

}//cierre de main
```