



Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

1. SENTENCIA SWITCH CASE

La estructura de control de selección **switch** en C permite seleccionar una entre múltiples alternativas. Es especialmente útil cuando la selección se basa en valores exactos de una variable. Escoger alternativas de valores exactos es más sencillo con un **switch** que con estructuras **if** anidadas.

Sintaxis de la instrucción:

```
switch (valor)
{
    case 1: ejecuta esto si valor es 1
        break;
    case 2: ejecuta esto si valor es 2
        break;
    case 3: ejecuta esto si valor es 3
        break;
    default: ejecuta esto si valor no es ninguno de
        los anteriores
}
```

Figura 1. Sintaxis del Switch Case

Se evalúa la expresión de control o selector (en este caso valor) y se compara con cada una de las etiquetas **case** para ejecutar la secuencia de acciones hasta la sentencia **break**. Cada etiqueta es un valor único y diferente de los otros, la etiqueta **default**, es opcional y se utiliza cuando se sospecha que pueden venir valores diferentes a los de las etiquetas. Cuando el computador encuentra la sentencia **break**, saltará a la primera sentencia luego de la llave de cierre del **switch**.

La variable que se utilice para el switch, se denomina expresión de control o selector, esta solo puede ser de tipo **int** o **char**.



Universidad de Pamplona
Facultad de Ingenierías y Arquitectura
Ingeniería Electrónica
Fundamentos de Programación
Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

```
#include<stdio.h>

int main(){

    int hijos;

    printf("Ingrese la cantidad de hijos que usted tiene: \n");
    scanf("%i",&hijos);

    switch(hijos){
        case 0:
            printf("No le corresponde asignacion familiar por hijo");
            break;
        case 1:
            printf("Le corresponden $95000 por su unico hijo");
            break;
        case 2:
            printf("Le corresponden por sus dos hijos");
            break;
        default:
            printf("Le corresponden $200000 por tenes mas de 2 hijos");
    }

    return 0;
}
```

Figura 2. Ejemplo de Switch Case

Se define la variable del tipo entero hijos, se le pide al usuario que ingrese el número de hijos que posee y luego entra en acción la estructura switch, actuando de diferentes formas dependiendo del valor de la variable hijos en tiempo de ejecución.

Uno de los usos más frecuentes del switch está en la implementación de menús, esto es, presentar en pantalla una serie de alternativas para que el usuario elija una.

Ejemplo 2: Mostrar en pantalla un menú con tres opciones y la captura de la selección del usuario. El programa evaluará la opción y escogerá una alternativa; en caso de que la opción sea diferente de 1, 2 o 3, utilizará la opción por defecto. AREA FIGURAS



Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

ESTRUCTURAS REPETITIVAS

En la elaboración de programas es importante que existan instrucciones que permiten que el computador ejecute un grupo de instrucciones varias veces sin tener que volverlas a escribir, estas son: la instrucción **mientras** que y la instrucción **para**.

BUCLES E ITERACIONES

Un bucle o ciclo es una instrucción que permite realizar tareas repetitivas. Un bucle consta de tres partes fundamentales: DECISION (simple o compuesta), CUERPO DEL BUCLE y la SALIDA DEL BUCLE.

Una iteración es la ejecución de todas las instrucciones del ciclo. Los bucles pueden ser sencillos o anidados (ciclos dentro de otros ciclos).

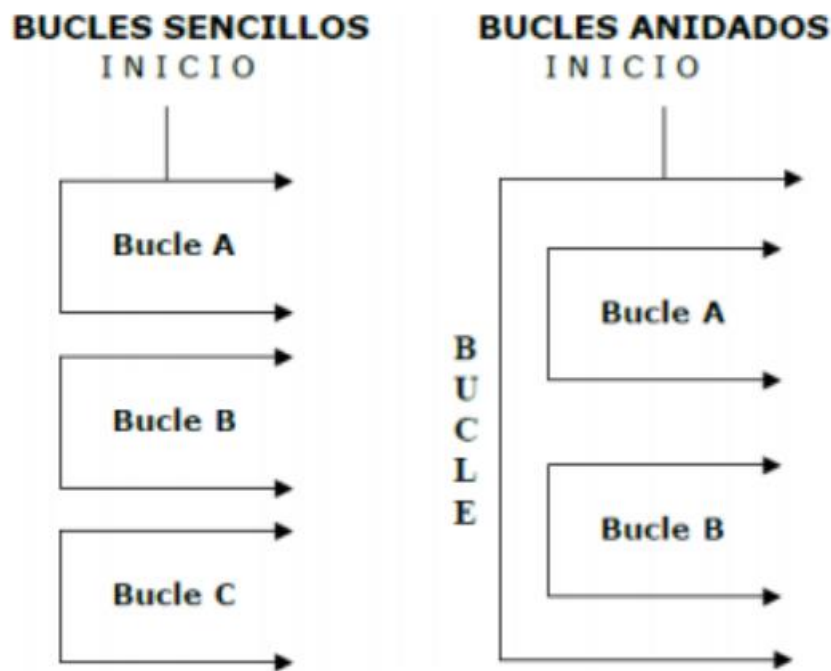


Figura 4. Bucles

Los procesos repetitivos son la base del uso del computador. En estos procesos se necesita normalmente contar los sucesos o acciones internas del bucle. Una forma de controlar un bucle es mediante un contador.



Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

Un contador es una variable cuyo valor se incrementa o disminuye en una cantidad constante por cada iteración.

El formato general es:

CONTADOR = CONTADOR + CANTIDAD_CONSTANTE

Pueden ser utilizados en vez de las expresiones: $x=x+1$ o $x+=1$ con tan solo escribir $x++$.
Un acumulador es una variable cuya misión es almacenar por cada iteración, cantidades variables resultantes de sumas o productos sucesivos. El formato general es:

ACUMULADOR = ACUMULADOR + CANTIDAD_VARIABLE

ACUMULADOR = ACUMULADOR * CANTIDAD_VARIABLE

En C se utiliza el operador de asignación que permite simplificar estas expresiones, así:

ACUMULADOR += CANTIDAD_VARIABLE

ACUMULADOR *= CANTIDAD_VARIABLE

OPERADORES DE INCREMENTO Y DECREMENTO

Operación	Sintaxis	Significado	Comentario
Postincremento	$i++$	$i = i + 1$	En una operación de asignación esta operación hace primero se realice la asignación y luego se incremente la variable en 1.
Preincremento	$++i$	$i = i + 1$	En una operación de asignación esta operación hace primero se realice la el incremento de la variable en 1 y luego se realice la asignación.
Postdecremento	$i--$	$i = i - 1$	En una operación de asignación esta operación hace primero se realice la asignación y luego se decremente en 1 la variable.
Predecremento	$--i$	$i = i - 1$	En una operación de asignación esta operación hace primero se realice la disminución de la variable en 1 y luego se realice la asignación.



Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

SINTAXIS DE LAS ESTRUCTURAS DE CONTROL REPETITIVAS EN C

C++, utiliza tres tipos de estructuras de control de repetición: **for**, **while** y **do-while**.

SENTENCIA **for:**

La estructura de control for (para), genera bucles controlados por contador, en los cuales se sabe de antemano cuantas son las iteraciones del bucle.

Sintaxis de la instrucción:

for (inicialización_contador; condición-expresión lógica; incremento o disminución)

```
{  
                                Bloque de sentencias  
}
```

Para controlar el número de veces que se repite el bucle, se requiere la utilización de un contador y especificar en qué valor va a iniciar, además en qué cantidad constante se incrementará o disminuirá esta variable. La condición es una expresión booleana que indica cuando se debe terminar con las iteraciones. Cada iteración en el ciclo produce el incremento (o disminución) del contador. Se acostumbra a utilizar los operadores incrementales o de disminución, para mostrar de una manera sencilla el cambio en el contador, así:

a++ cuando se desea incrementar en uno el contador.
a-- cuando se desea disminuir en uno el contador.
a+=2 cuando se desea incrementar en dos el contador.

El siguiente ejemplo:

```
int K;  
for (k=1; k <= 20; k+=2)  
    printf("k");
```

Los puntos y comas son indispensables para separar la inicialización del contador, con la expresión lógica y el incremento o disminución.

Imprime los números 1, 3, 5, 7, 9, 11, 13, 15, 17 y 19. La variable k es el contador e inicia en 1 y finaliza al llegar a 20, en la primera iteración imprimirá el valor 1. En la siguiente



Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

incrementa k en 2 y la variable cambia su valor a 3 y escribe en pantalla el número 3. En la siguiente aumenta k en 2 y la variable cambia su valor a 5, escribe en pantalla el número 5 y así sucesivamente hasta imprimir el número 19, cuando k toma el valor de 21, no ingresa al bucle por lo tanto este último dato no se imprime, el control del programa continuará con las instrucciones siguientes al bucle.

La declaración de la variable de control si se desea se puede hacer en la misma instrucción **for** así:

```
for (int k=1; k<=20; k+=2)
```

```
    printf("k");
```

Ejemplo: Sumar 10 números enteros introducidos por teclado

Variables: NU: Numero, SUMA: Acumulador de sumas, i: contador que controla el ciclo.

Pseudocódigo:

Inicio

SUMA ← 0

para i ← 0,10,1 **haga**

leer NU

 SUMA ← SUMA + NU

fin_para

escribir SUMA

fin



Estructuras cíclicas, For, While, Do While, Operadores de incremento y decremento, Switch Case

Ejercicios:

Fuente: <https://www.abrirllave.com/c/instruccion-repetitiva-for.php>

- Primeras diez potencias de 2
- Sucesión de números positivos y negativos
- Números pares del 1 al 30
- Números múltiplos de 4 entre el 64 y el 44

SENTENCIA while

La estructura de control **while** (mientras), es especialmente útil cuando se desea generar bucles controlados por centinela, esto es, cuando se sabe la condición de terminación del bucle, pero no se sabe de antemano cuantas son las iteraciones del bucle.

Sintaxis de la instrucción:

while (condición-expresión lógica)

```
{  
    Bloque de sentencias  
}
```

En este caso las sentencias se repiten, mientras la expresión lógica o condición del bucle sea verdadera.

La condición para ingresar al bucle se evalúa antes de que se ejecuten las sentencias. Si la condición es falsa, no se ingresará al bucle.

Note que no se coloca punto y coma (;) en la línea del while, de hacerse no ejecutará el bucle.