

Sentencias de Control

Programacion en C++

Taller de programación



Departamento de Informática
Universidad Técnica Federico Santa María

Operadores de comparación

Si $a = 10$, $b = 20$;

Operador	Comparación	Ejemplo	Resultado
<code>==</code>	Igualdad	<code>a == b?</code>	falso
<code>!=</code>	Desigualdad	<code>a != b?</code>	verdadero
<code>></code>	Mayor que	<code>a > b?</code>	falso
<code><</code>	Menor que	<code>a < b?</code>	verdadero
<code>>=</code>	Mayor o igual que	<code>a >= b?</code>	falso
<code><=</code>	Menor o igual que	<code>a <= b?</code>	verdadero

Ejemplos

Si $a = 2$, $b = 3$ y $c = 6$, ¿cuales de las siguientes expresiones son verdaderas? ¿cuales son falsas?

-
- 1: $(a == 5)$
 - 2: $(a * b >= c)$
 - 3: $(b + 4 > a * c)$
 - 4: $((b = 2) == a)$
-

Ejemplos

Si $a = 2$, $b = 3$ y $c = 6$, ¿cuales de las siguientes expresiones son verdaderas? ¿cuales son falsas?

-
- 1: $(a == 5)$
 - 2: $(a * b \geq c)$
 - 3: $(b + 4 > a * c)$
 - 4: $((b = 2) == a)$
-

Respuestas:

- 1. falsa, porque la variable a vale 2
- 2. verdadera, porque la expresión $a * b$ vale 6, lo que es mayor o igual al valor de c
- 3. falsa, porque $b + 4$ vale 7 y $a * c$ vale 12
- 4. verdadera, porque primero se asigna 2 a b , y se compara este nuevo valor de b con a

Operadores lógicos

Podemos usar los operadores lógicos negación (!), conjugación (&&) y disyunción (||) para combinar expresiones:

a	b	!a	a && b	a b
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false

Ejemplos:

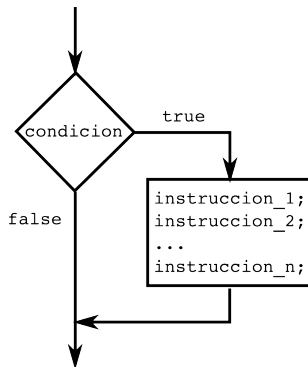
<code>((5 == 5) && (3 > 6))</code>	<i>// esta expresion es falsa (true && false)</i>
<code>((5 == 5) (3 > 6))</code>	<i>// esta expresion es verdadera (true false)</i>

- ▶ La mayoría de los programas incluyen la elección de un caso entre los posibles, en función del valor particular de alguno de los datos
- ▶ Existen dos sentencias de selección de casos en C++:
 - condicional simple: `if, if ...else`
 - condicional múltiple: `switch`

Selección: if

Si *condicion* es verdadera, se ejecutarán las instrucciones asociadas al *if*

```
if (condicion){  
    instruccion_1;  
    instruccion_2;  
    ...  
    instruccion_n;  
}
```



Ejemplo

Que hace el siguiente programa?

```
#include <iostream>
using namespace std;

int main()
{
    int a = 100;

    if (a == 100){
        cout << "El valor de a es";
        cout << a << endl;
    }

    return 0;
}
```

Ejemplo

Que hace el siguiente programa?

```
#include <iostream>
using namespace std;

int main()
{
    int a = 100;

    if (a == 100){
        cout << "El valor de a es";
        cout << a << endl;
    }

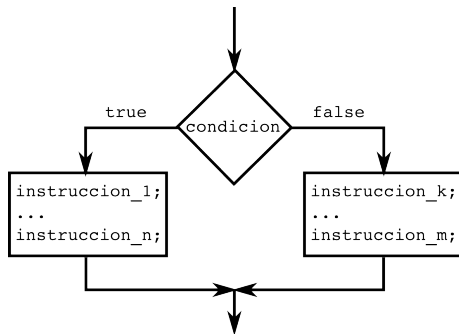
    return 0;
}
```

Resultado: imprime la linea El valor de x es 100

Selección: if ...else

Si *condicion* es verdadera, se ejecutarán las instrucciones asociadas al *if*, sino se ejecutan las instrucciones asociadas al *else*

```
if (condicion){  
    instruccion_1;  
    ...  
    instruccion_n;  
}  
else {  
    instruccion_k;  
    ...  
    instruccion_m;  
}
```



Ejemplo

Que hace el siguiente programa?

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;

    if (a > b){
        cout << "El valor de a es mayor que b";
    }
    else {
        cout << "El valor de b es mayor o igual que a";
    }

    return 0;
}
```

Ejemplo

Que hace el siguiente programa?

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;

    if (a > b){
        cout << "El valor de a es mayor que b";
    }
    else {
        cout << "El valor de b es mayor o igual que a";
    }

    return 0;
}
```

Resultado: indica cual de los dos valores ingresados es mayor

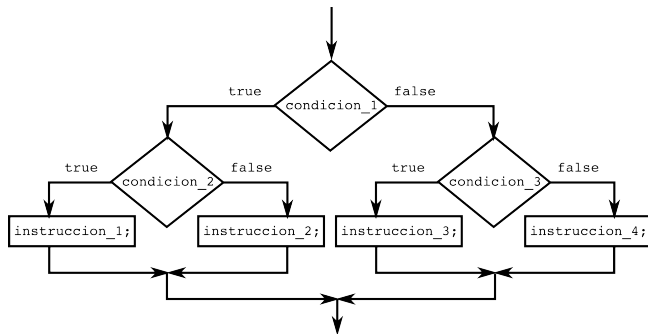
Ejercicios

- ▶ Desarrollar un programa que reciba un numero entero como entrada e indique si este es par.
- ▶ Desarrollar un programa que determine las soluciones para una ecuación cuadrática, el cual recibe como parámetros el valor de los coeficientes de la ecuación.
- ▶ Realizar un programa que solicite el ingreso de tres números enteros y muestre por pantalla los números ordenados de menor a mayor.

Selección: if ...else anidados

Pueden anidar los if ...else

```
if (condicion_1){  
    if (condicion_2){  
        instruccion_1;  
    }  
    else{  
        instruccion_2;  
    }  
}  
else {  
    if (condicion_3){  
        instruccion_3;  
    }  
    else{  
        instruccion_4;  
    }  
}
```



Ejercicio

Un año es bisiesto si es divisible por 4, excepto si es divisible por 100 y no por 400. Escriba un programa que indique si un año es bisiesto.

Casos de prueba:

Ingrese un año: **1988**

1988 es bisiesto

Ingrese un año: **2011**

2011 no es bisiesto

Ingrese un año: **1700**

1700 no es bisiesto

Ingrese un año: **2400**

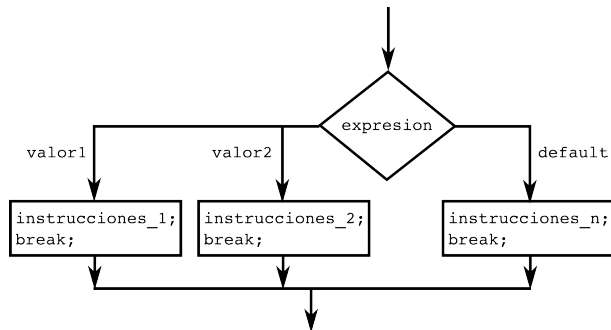
2400 es bisiesto

Selección: `switch`

- ▶ Los “if anidados” plantean la problemática de bifurcación múltiple
- ▶ Puede llegar a ser difícil de leer
- ▶ C++ proporciona una estructura alternativa mas legible para aquellos casos en los que la bifurcación múltiple se realiza mediante comparación de enteros, denominada `switch`.

Selección: switch

```
switch (expresion) {  
  case valor1:  
    instrucciones_1;  
    break;  
  case valor2:  
    instrucciones_2;  
    break;  
  ...  
  default:  
    instrucciones_n;  
    break;  
}
```



El fragmento etiquetado con default es opcional.

Ejemplo

```
#include <iostream>
using namespace std;

int main()
{
    int a;
    cout << "Ingresa un numero: ";
    cin >> a;

    switch ( a ) {
        case 10:
            cout << "hola\n";
            break;
        case 20:
            cout << "chao\n";
            break;
        default:
            cout << "hola y chao\n";
            break;
    }
    return 0;
}
```

Implementar una calculadora que permita:

- ▶ Sumar
- ▶ Restar
- ▶ Multiplicar
- ▶ Ordenar de menor a mayor 3 números y viceversa.

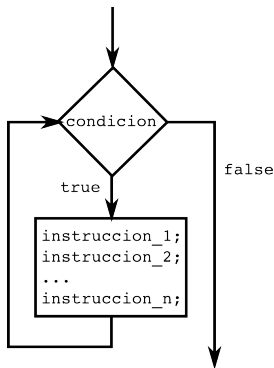
Repetición

- ▶ Las estructuras de control repetitivas, como su nombre lo indica, repiten la ejecución de un grupo de instrucciones una determinada cantidad de veces
- ▶ En C++, las estructuras de control de este tipo son:
 - `while`
 - `for`

Repetición: while

Mientras la condición sea verdadera, la(s) instrucción(es) se repite(n)

```
while (condicion) {  
    instruccion_1;  
    instruccion_2;  
    ...  
    instruccion_n;  
}
```



Dentro de las sentencias del cuerpo del while, se debe modificar alguna variable que se encuentre en la condición, sino se puede producir un ciclo infinito

Ejemplos

- El siguiente ejemplo produce un ciclo infinito

```
int variable = 1;
while (variable != 10){
    cout << variable << " ";
}
```

- Este otro ejemplo no

```
int variable = 1;
while (variable != 10){
    cout << variable << " ";
    variable++;
}
```

Ejercicio

Escriban un programa que muestre todas las potencias de 2 menores a 20000.

El resultado debe estar formateado de la siguiente manera:

- ▶ 5 números por línea
- ▶ Los números deben estar separados por comas

La salida debe lucir así:

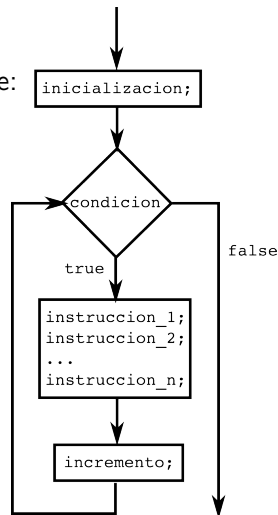
```
1, 2, 4, 8, 16,  
32, 64, 128, 256, 512,  
1024, 2048, 4096, 8192, 16384
```

Repetición: for

El for se usa cuando se conocen los límites del bucle:

- ▶ su parámetro de inicialización,
- ▶ su criterio de terminación, y
- ▶ su instrucción de incremento.

```
for (inicializacion; condicion; incremento){  
    instruccion_1;  
    instruccion_2;  
    ...  
    instruccion_n;  
}
```



Repetición: for

```
for (inicializacion; condicion; incremento){  
    instruccion_1;  
    instruccion_2;  
    ...  
    instruccion_n;  
}
```

- ▶ `inicializacion` es la instrucción que inicializa la iteración (se ejecuta una vez al iniciar).
- ▶ `condicion` es la expresión que determina cuando se termina el `for`. Esta expresión se evalúa al principio de cada iteración, y cuando la expresión llega a ser falsa, finaliza la ejecución del `for`.
- ▶ `incremento` es una instrucción que se invoca en cada iteración del bucle.

Ejemplos

- El siguiente trozo de código calcula $\sum_{i=0}^{100} i = 1 + 2 + \dots + 99 + 100$

```
int resultado = 0;
for(i = 0; i <= 100; i++){
    resultado = resultado + i;
}
```

- Lo mismo, pero al revés

```
int resultado = 0;
for(i = 100; i >= 0; i--){
    resultado = resultado + i;
}
```

Ejercicios

- ▶ Desarrolle un programa que muestre todos los años bisiestos del siglo XXI.
- ▶ Desarrolle un programa que escriba una tabla de dos columnas para la conversión entre las temperaturas en grados Fahrenheit, comprendidas entre 0 y 300, según incrementos de 1, y su equivalente en grados centígrados.

La conversión esta dada por la formula $C = \frac{5 \times (F - 32)}{9}$, donde C esta en centígrados y F en Fahrenheit.