



Programación en C++

MSc. Víctor Melchor Espinoza
cpp.ctic@gmail.com

Unidad



Sesión 4: Estructuras de Iteración

Contenidos

- a) Ciclo do - while.
- b) Ejemplos de Validación

Aprendizajes esperados

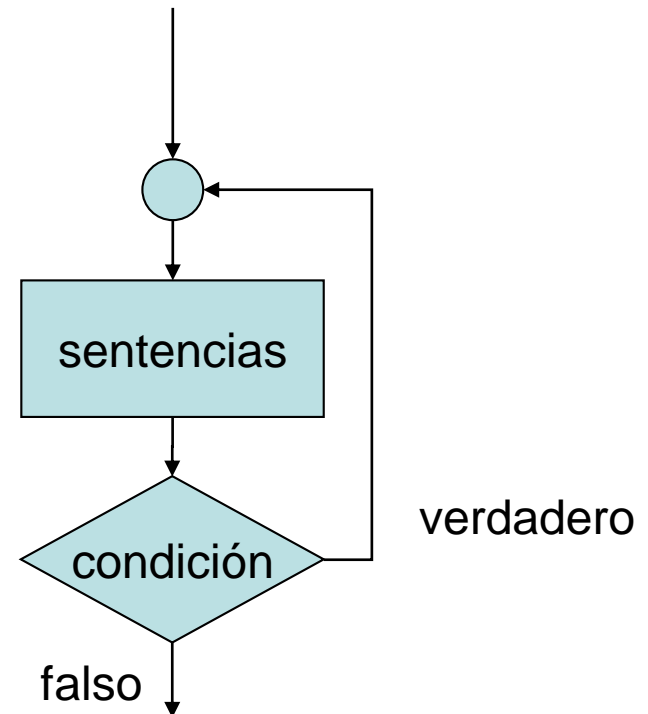
- Comprende el uso de la estructura do - while

Ciclo do - while

El ciclo do-while es similar al ciclo while excepto que la prueba se realiza al final del ciclo, esto fuerza a que se ejecute por lo menos una vez.

Sintaxis

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



Lectura con validación

El ciclo do-while puede usarse para validar la entrada de datos.

Algoritmo de validación:

1. Hacer
2. Leer datos
3. Mientras datos inválidos

Ejemplo de validación

```
#include <iostream.h>
```

```
main() {  
    int a,b;  
    do{  
        cout << "Teclee un número positivo para A: ";  
        cin >> a;  
    }while(a<=0) ;  
    do{  
        cout << "Teclee un número positivo para B: ";  
        cin >> b;  
    }while(b<=0) ;  
    cout <<"a * b = " << a*b << endl;  
    system("pause") ;  
}
```

Validación de una fecha

Para validar una fecha debemos asegurar que los días caigan en el rango de 1 a 31 y los meses de 1 a 12.

Una mejora consiste en verificar los días de acuerdo al mes.

Para febrero en año bisiesto el máximo de días es 29.

Para verificar año bisiesto usamos

```
anyoBisiesto = (a%4==0) && (a%100!=0) ;
```

Para los meses de 30 y 31 días:

```
mes30dias = (m==4 || m==6 || m==9 || m==11) ;
```

```
mes31dias = (m==1 || m==3 || m==5 || m==7 ||  
m==8 || m==10 || m==12) ;
```

Expresión para checar fecha:

```
fechaValida = (d<=31 && mes31dias) || (d<=30 &&  
mes30dias) || (d<=28 && m==2) || (d<=29 && anyoBisiesto) ;
```

Programa

```
#include <iostream.h>
```

```
main() {
```

```
    int d,m,a;//día, mes, año
```

```
    int anyoBisiesto,mes30dias,mes31dias,fechaValida;
```

```
    do{
```

```
        cout << "Teclee una fecha (dd mm aa): ";
```

```
        cin >> d >> m >> a;
```

```
        anyoBisiesto = (a%4==0) && (a%100!=0);
```

```
        mes30dias = (m==4 || m==6 || m==9 || m==11);
```

```
        mes31dias = (m==1 || m==3 || m==5 || m==7
```

```
        || m==8 || m==10 || m==12);
```

```
        fechaValida = (d<=31 && mes31dias) || (d<=30 &&
```

```
        mes30dias) || (d<=28 && m==2) || (d<=29 &&
```

```
        anyoBisiesto);
```

```
    }while(!fechaValida);
```

```
cout << d << " de ";  
    switch(m) {  
        case 1:cout << "enero";break;  
        case 2:cout << "febrero";break;  
        case 3:cout << "marzo";break;  
        case 4:cout << "abril";break;  
        case 5:cout << "mayo";break;  
        case 6:cout << "junio";break;  
        case 7:cout << "julio";break;  
        case 8:cout << "agosto";break;  
        case 9:cout << "septiembre";break;  
        case 10:cout << "octubre";break;  
        case 11:cout << "noviembre";break;  
        case 12:cout << "diciembre";break;  
    }  
    cout << " de " << a << endl;  
    system("pause");  
}
```


tarea

Escriba un programa que lea tres números enteros positivos, el tercero de los cuales deberá ser mayor que los otros dos y menor o igual a la suma de los dos primeros.

Ciclos controlados por centinela

Si no se conoce de antemano el número de datos, se utiliza un valor de entrada especial como una bandera o centinela para terminar la entrada de datos.

En el caso del promedio el centinela puede ser -1.

Algoritmo

1. Solicitar un nuevo valor para promediar
2. Mientras valor diferente de -1 hacer
3. Acumular suma e incrementar contador
4. Solicitar nuevo valor para promediar
5. Fin ciclo
- 6 si contador > 0
7. Calcular promedio e imprimir

ciclo controlado por centinela

Cálculo de el promedio de N (desconocido) números. Se utiliza un valor especial para detener la entrada de datos.

```
int main() {
    float suma = 0.0, num, promedio;
    int contador = 0;
    cout << "Teclee un número (-1 = fin):";
    cin >> num;
    while(num != -1) {
        suma = suma + num;
        contador = contador + 1;
        cout << "Teclee un número (-1 = fin):";
        cin >> num;
    }
    if(contador>0) {
        promedio = suma/contador;
        cout << "\nEl promedio es: " << promedio << endl;
    }
    else
        cout << "\nNo se teclearon valores" << endl;
    getch();
}
```

Ejemplo

Se desea un programa para leer las calificaciones de un grupo de alumnos y calcular el promedio general, así como el número de alumnos aprobados y reprobados.

No conocemos el número de alumno, por tanto es adecuado el esquema del centinela.

Primer esbozo de algoritmo

Leer siguiente calificación

Mientras calificación diferente de -1

acumular suma

si calificación \geq 6

incrementar aprobados

sino

incrementar reprobados

Incrementar total de alumnos

Leer siguiente calificación

Fin ciclo

Calcular promedio

Imprimir resultados

Variables:

contador – contador de calificaciones

aprobados – contador de aprobados

reprobados – contador de reprobados

cal – calificación

suma – suma de calidficaciones

prom – promedio de calificaciones

Iniciación

```
int total = 0, aprobados = 0, reprobados = 0;
```

```
float cal, suma = 0, prom;
```

Lectura de calificación

```
cout << "teclee calificacion (-1 para terminar):";  
cin >> cal;
```

Ciclo de lectura

```
while(cal != -1) {  
    suma += cal;  
    if(cal >= 6)  
        aprobados++;  
    else  
        reprobados++;  
    total++;  
    cout << "teclee calificacion (-1 para terminar):";  
    cin >> cal;  
}
```

Cálculo del promedio:

```
prom = suma/total;
```

Impresión de resultados:

```
cout << "Total de alumnos: " << total << endl;
```

```
cout << "Promedio: " << prom << endl;
```

```
cout << "# de aprobados: " << aprobados << endl;
```

```
cout << "# de reprobados: " << reprobados << endl;
```

ejemplo

Se desea un programa que funcione como una calculadora.

El programa aceptará expresiones de la forma:

numero op numero

Donde op puede ser: +, -, *, / y ^ (potencia)

Ejem.: 5.7+4.2, 6.7/2.5,

El programa terminará cuando se teclee la secuencia 0x0, donde x es cualquier carácter.

El programa puede hacerse con un esquema de ciclo con centinela.

Algoritmo

1. Iniciar programa
2. Leer siguiente expresión
3. Mientras expresión diferente de "0*0" hacer
4. Calcular e imprimir resultado
5. Leer siguiente expresión
6. Fin ciclo

El paso 4 sería

4. en caso de que op sea
 - 4.1. '+' sumar los dos números
 - 4.2. '-' restar los dos números
 - 4.3. '*' multiplicar los dos números
 - 4.4. '/' dividir los dos números
 - 4.5. '^' potencia de los dos números

Programa

```
#include <iostream.h>
#include <math.h>
main() {
    float x,y,r;//datos de entrada y resultado
    char op;//operador
    cout <<">";//indicador
    cin >> x>> op >> y;//lectura de datos
    while (!(x==0&&y==0)){
        switch(op) {
            case '+':r = x+y;break;
            case '-':r = x-y;break;
            case '*':r = x*y;break;
            case '/':r = x/y;break;
            case '^':r = pow(x,y);break;
            default:r=0;
        }
        cout << r << endl;
        cout <<">";
        cin >> x>> op >> y;
    }
}
```

ciclo do-while controlado por centinela

Cálculo de el promedio de N números. Se utiliza un valor especial para detener la entrada de datos.

```
int main() {
    float suma = 0.0, num, promedio;
    int contador = 0;
    do{
        cout << "Teclee un número (-1 = fin):";
        cin >> num;
        if(num != -1){
            suma = suma + num;
            contador = contador + 1;
        }
    }while(num != -1);
    if(contador>0){
        promedio = suma/contador;
        cout << "\nEl promedio es: " << promedio << endl;
    }
    else
        cout << "\nNo se teclearon valores" << endl;
    getch();
}
```

Determinar los primos de 1 a 100

```
int main() {
    int n, num = 2, esPrimo, limite;
    while(num <= 100) {
        esPrimo = 1;
        n = 2;
        limite = (int)sqrt(num);
        while(n <= limite && esPrimo) {
            if(num % n == 0) //es divisible entre n
                esPrimo = 0;
            n++;
        }
        if(esPrimo)
            cout << num << " ";
        num++;
    }
    getch();
    return 0;
}
```

Diagram illustrating the loops in the code:

- lazo externo** (outer loop): Indicated by a large curly brace on the left, it encompasses the entire `while(num <= 100)` loop.
- lazo interno** (inner loop): Indicated by a smaller curly brace on the left, it encompasses the nested `while(n <= limite && esPrimo)` loop.

Tarea casera

Un número es perfecto si es igual a la suma de sus divisores, por ejemplo 6 es perfecto porque $6 = 1 + 2 + 3$. Escriba un programa para encontrar todos los números perfectos entre 1 y 10000.

Tarea casera

Escriba programa con un ciclo controlado por centinela para leer una serie de números hasta que el usuario teclee el número 9999. Dentro del ciclo cuente los múltiplos de 2, de 3 y de 5. Imprima cuantos múltiplos se teclearon en cada caso.

Ejemplo:

Teclee un número: 6

Múltiplos de 3: 3

Múltiplos de 5: 2

Teclee un número: 9

Teclee un número: 15

Teclee un número: 20

Teclee un número: 8

Teclee un número: 9999

Múltiplos de 2: 3

Ciclos anidados

Muchos algoritmos se definen mediante ciclos dentro de otros ciclos.

Es posible incluir dentro de un ciclo cualquier estructura de control incluyendo otro ciclo.

Ejemplo

Programa para imprimir un rectángulo de asteriscos (*) leyendo el tamaño del lado y la altura con un máximo de 20.

Ejemplo:

Si se teclea 10 8 , imprimirá

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```


Este problema puede resolverse fácilmente haciendo un ciclo dentro de otro. El algoritmo es

1. Leer el valor de n y m
2. Si $n > 1$ y $n < 20$ y $m > 1$ y $m < 20$
3. $r = 0$
4. Mientras $r < m$ hacer
5. $c = 0$
6. Mientras $c < n$ hacer
7. Imprimir “*”
8. Incrementar c
9. Fin ciclo
10. Cambiar de línea
11. Incrementar r
12. Fin ciclo

```
#include <iostream.h>
#include <iomanip.h>
main() {
    int n,m,c,r;
    cout << "ancho del rectangulo (maximo 20):";
    cin >> n;
    cout << "alto del rectangulo (maximo 20):";
    cin >> m;
    if(n>1 && n<20&&m>1&&m<20) {
        r=0;
        while(r<m) {
            c = 0;
            while(c<n) {
                cout << '*';
                c++;
            }
            cout << endl;
            r++;
        }
    }
    cout<<endl;
    system("PAUSE");
}
```

Tarea casera

Escriba un programa para dibujar un patrón de tablero de ajedrez de tamaño 2 a 20 como se muestra:

```
*  *  *  *  
  *  *  *  *  
*  *  *  *  
  *  *  *  *
```

Utilice ciclos anidados y sentencias de salida que impriman un asterisco y espacio en blanco.

Ayuda: note que las líneas impares comienzan con asterisco y las pares con espacio.

Operadores de asignación

En C existen operadores para abreviar las operaciones de asignación.

Por ejemplo: `c = c + 3` puede escribirse como `c += 3`.

En general

`variable = variable operador expresión`

es equivalente a

`variable operador= expresión`

Ojo

`a *= c + d`

equivale a

`a = a * (c + d)`

no

`a = a * c + d`

Ejemplos

`a = a + 3;`

`a += 3`

`c = 2*c;`

`c *= 2;`

`x = x - z;`

`x -= z;`

`s = s*b - s*c;`

`s *= b - c;`

`d = d/(f + 5);`

`d /= f + 5;`

`r = r % 5;`

`r %= 5;`

Operadores de Incremento y decremento

Operador	Llamado	Ejemplo	Descripción
++	preincremento	++a	Incrementa a en 1, luego utiliza el nuevo valor en la expresión en que reside a.
++	posincremento	a++	Toma el valor actual de a para la expresión en que reside a, luego incrementa a.
--	predecremento	--a	Decrementa a en 1, luego utiliza el nuevo valor en la expresión en que reside a.
--	posdecremento	a--	Toma el valor actual de a para la expresión en que reside a, luego decrementa a.

Números primos

Un número primo es aquel que solo es divisible entre si mismo y entre 1. Por ejemplo: 1, 2, 5, 47.

Para saber si un número es primo debemos obtener el residuo de la división del número entre todos los números desde 2 hasta su raíz cuadrada. Si el residuo es diferente de cero, el número es primo.

Algoritmo

- | | |
|----------------------------------|------------------------------|
| 1. Leer el numero | 8. div++ |
| 2. div = 2; | 9. finlazo |
| 3. esPrimo = 1 | 10. si esPrimo |
| 4. limite = raiz de numero | 11 imprime "primo" |
| 5. mientras div<limite o esPrimo | 12 sino |
| 6. si numero%div == 0 | 13 imprime "NO primo" |
| 7. esPrimo = 0 | |

Determinación de número primo

```
int main(){
    int div = 2, num, esPrimo = 1, limite;

    cout << "Teclee un número > 2: ";
    cin >> num;
    limite = (int)sqrt(num);
    while(div <= limite && esPrimo){
        if(num % div == 0)    //es divisible entre n
            esPrimo = 0;
        div++;
    }
    if(esPrimo)
        cout << "\nEl número " << num << " es primo\n";
    else
        cout << "\nEl número " << num << " NO es primo\n";
    getch();
    return 0;
}
```


Tarea

¿Cuál es la salida de los siguientes ciclos?

```
i = 0;
```

```
while (i < 10)
```

```
    cout << 2*i << setw(10) << 3*i*i;
```

```
    i += 3;
```

```
}
```

```
i = 5;
```

```
while (i > -20)
```

```
    cout << 2*i << setw(10) << 2*i+1;
```

```
    i -= 3;
```

```
}
```

Tarea casera

Escriba un programa para encontrar todos los divisores de un número.

Ejemplo:

Teclee un número: 246

Los divisores de 246 son: 2, 3, 6, 41, 82, 123