

PROGRAMACIÓN I

C++

2021-22



UCM

Grado en Estadística Aplicada. EUE.



2

Tema 2.- Programación estructurada

Transparencias originales de Isabel Riomoros

Programación estructurada

3

T

E

M

A

3

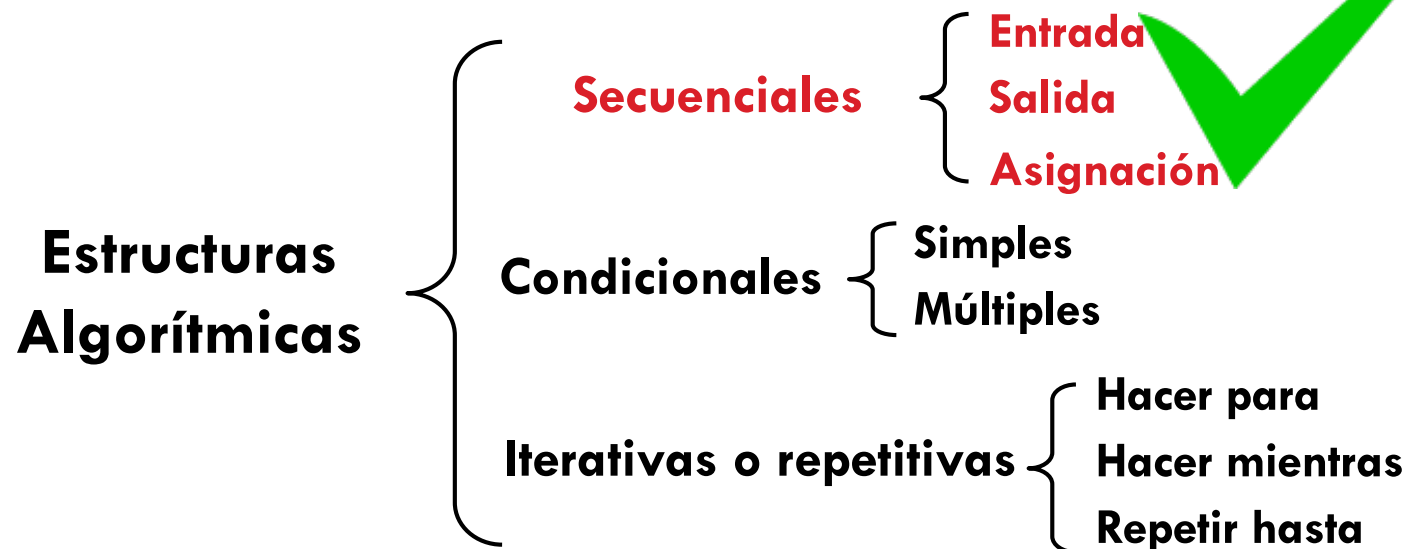
- **Sentencias condicionales**
- **Sentencias iterativas o repetitivas**



Introducción

4

programas lineales - comienzan por la primera Sentencia y acababan por la última, ejecutándose todas una sola vez.



Introducción

5

cout ,cin, asignación(=)

- Sólo con estas sentencias podemos resolver problemas sencillos.

Sentencias IF y SWITCH

- Cuando nos interesa que dependiendo de los valores de los datos, se ejecuten unas sentencias u otras → **sentencias condicionales**.

Sentencias WHILE, DO-WHILE y FOR

- A veces nos interesará repetir una sentencia ó sentencias un número determinado de veces → **sentencias de control iterativas ó repetitivas** (ciclos o bucles).



C++

6

Sentencias condicionales

IF, IF-ELSE, SWITCH



Sentencia condicional simple: IF

7

El formato general de una sentencia **if** es la siguiente:

```
if (condición) {  
    Sentencia 1;  
    ...  
    Sentencia n;  
}
```

```
if (condición)  
    Sentencia;
```

!!!OJO!!!

Si se cumple la condición, entonces se ejecuta la *Sentencia* ó el *bloque de Sentencias*; en caso contrario, no se ejecutan.



C++

Sentencia condicional simple: IF

8

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

```
int main(){
    int a;
    cin >> a;

    if (a<0)
        a = -a;

    cout << a;
    return 0;
}
```

Valor absoluto

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

```
int main(){
    ...

    if (cantidad > 3){
        descuento = 0.2;
        precio = n*descuento;
    }

    ...
    return 0;
}
```

Descuento en un producto si
compras más de tres unidades



C++

Sentencia condicional doble : IF-ELSE

9

```
if (condición) {  
    Sentencias 1;  
}  
else {  
    Sentencias 2;  
}
```

```
if (condición)  
    Sentencia 1;  
else  
    Sentencia 2;
```

Si se cumple la condición, se ejecutan las Sentencias del primer bloque;

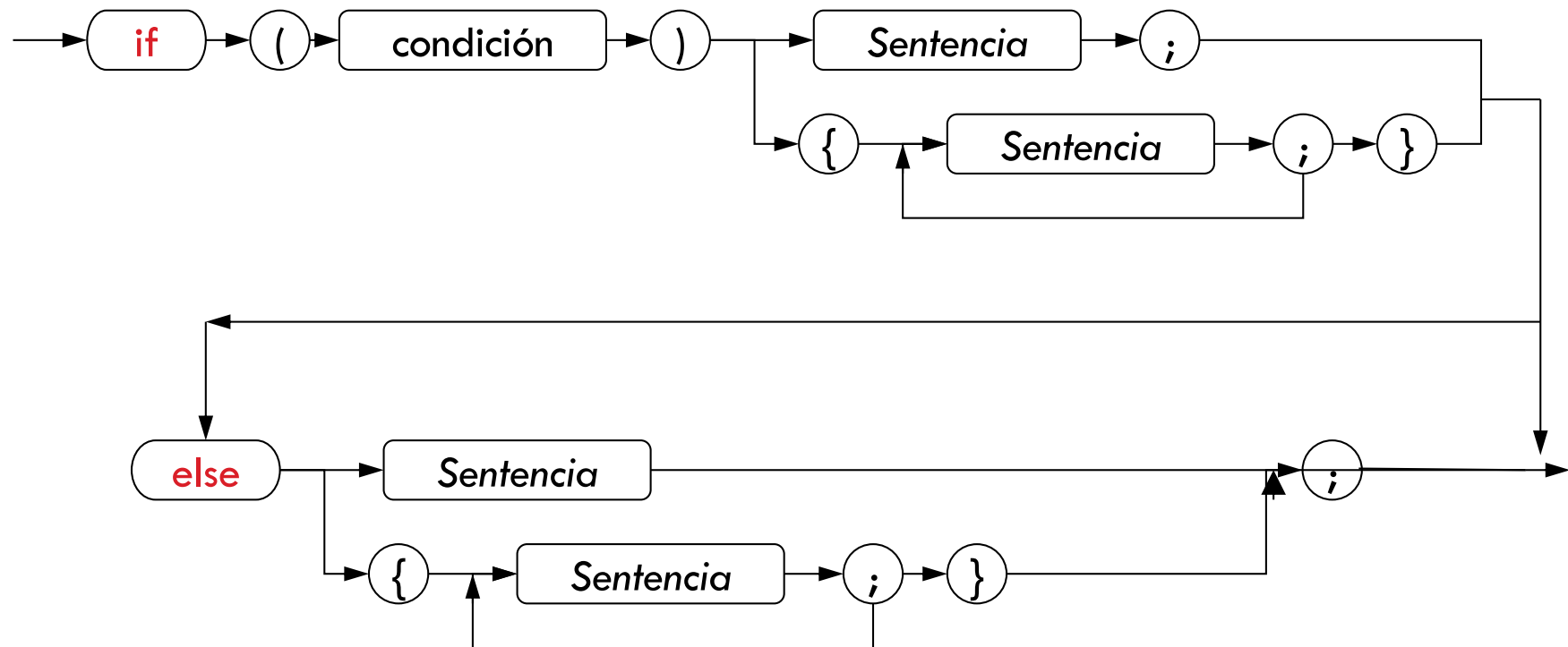
si no, se ejecutan las Sentencias del segundo bloque.



Sentencia condicional IF, IF-ELSE

10

Diagrama sintáctico:



C++

Sentencia condicional doble : IF-ELSE

11

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

```
int main(){
    int a, x;
    cin >> a;

    if (a==0)
        x = 1;
    else
        x= 0;

    cout << x;
    return 0;
}
```

Dado un número a, si es igual a cero asignar 1 a la variable x, sino asignar 0 a x

```
#include <iostream>
using namespace std ;
int main ( ) {
    const char VOCAL = 'e' ;
    char letra ;
    cout << "Introduzca una vocal : " ;
    cin >> letra;
    if ( letra == VOCAL) {
        cout << "Has acertado la vocal, enhorabuena!!!\n";
    }
    else {
        cout << "Ohhhhh no has acertado\n" ;
    }
    return 0 ;
}
```

Leer una letra del input y comprobar si he acertado con la letra constante que tenía definida



Sentencia condicional doble : IF-ELSE anidadas

12

Sentencias IF-ELSE anidadas

```
if (condición1)
    Sentencia 1;
else
    if (condición2)
        Sentencia 2;
    else
        Sentencia 3;
```

```
if (condición1)
    Sentencia 1;
else
    if (condición2)
        Sentencia 2;
    else
        if (condición3)
            Sentencia 3;
        else
            Sentencia 4;
```



Ejemplo

13

```
#include <iostream>
```

```
using namespace std ;
```

```
int main ( ) {
```

```
    double nota ;
```

```
    cout << " Introduzca la nota : " ;
```

```
    cin >> nota ;
```

```
    if (nota < 0 || nota > 10 )
```

```
        cout << "Nota no valida \n" ;
```

```
    else if (nota < 5.0)
```

```
        cout << " Suspenso\n" ;
```

```
    else if (nota < 7.0)
```

```
        cout << "Aprobado\n" ;
```

```
    else if (nota < 9.0)
```

```
        cout << "Notable \n" ;
```

```
    else cout << " Sobresaliente \n" ;
```

```
    return 0 ;
```

```
}
```

Dado un número real que denota una posible nota de una asignatura indicar si es aprobado, suspenso, notable, sobresaliente



Ejercicios

14

```
if(b!=0) if (b+3>a) sentencia0;  
else if(a+b<c) sentencia1; else if (b<10) sentencia2;  
else {sentencia3; if(a-b>x) sentencia4; } else sentencia5;
```

```
If (a>=0)  
if (a<=5)  
cout<<"a está entre 0 y 5";  
else cout<<"a debe ser menor que cero";
```

Si a toma un valor menor que 0 ¿qué pasará?



Sentencias Condicionales

15

```
#include <iostream>
```

```
int main(){
```

```
    int a, b, c, max;
```

```
    cin >> a ;
```

```
    cin >> b;
```

```
    cin >> c;
```

```
    if (a > b)
```

```
        if (a > c)
```

```
            cout << a;
```

```
        else
```

```
            cout << c;
```

```
    else
```

```
        if (b > c)
```

```
            cout << b;
```

```
        else
```

```
            cout << c;
```

```
    return 0;
```

```
}
```

¿Qué problema
resuelve este
código
?



C++

Errores comunes (sintaxis correcta, lógica no)

16

- ❑ Escribimos esto cuando queremos comprobar que $x=3$, utilizamos el operador de asignación ($=$) en vez de utilizar el operador de igualdad ($==$)
- ❑ El efecto de esta expresión es asignar el valor 3 a la variable x y producir como resultado de la expresión el valor asignado, o sea, 3. Como el valor es distinto de cero el resultado de la expresión es verdadero.

`if (x=3)`

`x==3`

`if (0<= x<= 3)`

- ❑ Se evalúa la expresión $0 \leq x$ que produce un valor de verdadero o falso en función del valor de x
- ❑ Luego se compara ese valor que es 0 ó 1 con el valor 3, siempre va a ser menor, por lo que la expresión **$0 \leq x \leq 3$** siempre va a ser cierta

`0<= x && x <= 3`

C++



Otro error (sintaxis correcta, lógica no)

17

```
if ( respuesta == s || respuesta == S )  
    cout << " Afirmativo";
```

- se compara el valor de la variable respuesta, que ha sido introducida por el usuario, con los valores de las variables s y S que no se han iniciado a ningún valor

```
if ( respuesta == 's' || respuesta == 'S' )  
    cout << " Afirmativo";
```



Ejercicios

18

- ❑ Valor absoluto de un número entero
- ❑ Dados dos valores x e y , si $x > y$ sumarlos sino multiplicarlos
- ❑ Dado el sueldo de un trabajador, aplicar un aumento del 20 % si el sueldo es inferior a 600 euros.
- ❑ Determina si un entero es par y en caso de no serlo, determinar si es divisible por 3
- ❑ Dados los goles del equipo local (golLocal) y los goles del equipo visitante (golvisita), devuelva una cadena de caracteres indicando qué equipo ha ganado (local, visitante, empate).
- ❑ Dado un número de mes escribir el nombre
- ❑ En una tienda hacen un 30% de descuento a los clientes cuya compra supere los 60 euros ¿ Cual será la cantidad que pagará una persona por su compra?(dos formas)
- ❑ Ordenar tres números de forma ascendente
- ❑ Dados tres números encontrar el mayor de los tres
- ❑ Acaban de abrir una tienda en mi barrio y para promocionarla hace descuentos sobre el valor de la compra total según el número que saque de una bolsa, dicha bolsa contiene sólo 5 bolas numeradas del 0 al 4. Si el número es el 0 no se le hará ningún descuento, si es 1 se le hará un 10% de descuento, si es el 2 un 25%, si es el 3 un 50% y si es el 4 un 100%. Calcular la cantidad final que el cliente deberá pagar por su compra.



Calcular el área de diferentes polígonos

19

¿Qué área quieres calcular?

- 1. cuadrado**
- 2. rectángulo**
- 3. Triángulo**
- 4. Fin**



Sentencia condicional múltiple : SWITCH

20

sentencia witch

```
switch (selector)
{
    case constante1:
        Sentencia1 ó bloque de Sentencias
        break;
    case constante2:
        Sentencia2 ó bloque de Sentencias
        break;
    default:
        Sentencia2 ó bloque de Sentencias
}
```

Permiten comparar una 'variable' con distintos valores posibles, ejecutando para cada caso una serie de Sentencias específicas.



Sentencia condicional múltiple : SWITCH

21

sentencia switch

```
switch (selector) {  
    case constante1:  
        Sentencia1 ó bloque de Sentencias  
        break;  
    case constante2:  
        Sentencia2 ó bloque de Sentencias  
        break;  
    default:  
        Sentencia3 ó bloque de Sentencias  
}
```

El valor de **selector** debe ser un **número entero**. Puede ser una variable, una expresión o una llamada a una función.

Cada caso comienza con un **case** y termina con un **break**

¿Qué ocurre si se me olvida algún **break** ?



C++

Sentencia condicional múltiple : SWITCH

22

Ejemplo

```
#include <iostream>
int main () {
    int num;
    cin >> num;
    switch (num)
    {
        case 1:
            cout << "Ha introducido el nº 1\n";
        case 2:
            cout << "Ha introducido el nº 2\n";
            break;
        default:
            cout << "Ha introducido otro nº";
    }
    return 0;
}
```

Si al ejecutar el programa introducimos un 2, obtenemos el mensaje:

'Ha introducido el nº 2'

Si al ejecutar el programa introducimos un 1, obtenemos el mensaje:

'Ha introducido el nº 1'

'Ha introducido el nº 2'



Dada una nota escribir la calificación

23

```
#include<iostream>
using namespace std;
int main() {
    int nota;
    cout<<" Dame una nota: ";
    cin>>nota;
    switch(nota) {
        case 0;
        case 1;;
        case 2;;
        case 3;
        case 4:   cout<<"\nSuspenso"; break;
        case 5:   cout<<"\nAprobado"; break;
        case 6:   cout<<"\nBien"; break;
        case 7;;
        case 8:   cout<<"\nNotable"; break;
        case 9;;
        case 10:  cout<<"\nSobresaliente"; break;
        default: cout<<"esa nota es incorrecta";
    }
    return 0;
}
```



Pasar de euros a pts o de Pts a euros

24

.....

```
const float EURO= 166.386;
```

```
float n,x; int opcion;
```

```
cout<<"la cantidad: ";
```

```
cin>>n;
```

```
cout<<"1-Ptas a Euros 2-Euros a ptas";
```

```
cin>>opcion;
```

```
switch(opcion) {
```

```
    case 1:
```

```
        x=n/EURO;
```

```
        cout<<n<<" Pesetas son "<< x<<" Euros"; break;
```

```
    case 2:
```

```
        x=n*EURO;
```

```
        cout<<n<<" Euros son "<<x<<" Pesetas"; break;
```

```
    default: cout<<"incorrecta";
```

```
}
```

.....



Ejercicios

25

Programa que simule el funcionamiento de una calculadora que puede realizar las cuatro operaciones aritméticas básicas (suma, resta, producto y división) con valores numéricos enteros. El usuario debe especificar la operación con el primer carácter del primer parámetro de la línea de comandos: S o s para la suma, R o r para la resta, P, p, M o m para el producto y D o d para la división. Solicitar al usuario los dos enteros y la operación a realizar.



C++

Sentencias repetitivas o iterativas

WHILE, DO-WHILE, FOR



Sentencias Iterativas o repetitivas

27

- Controlan la repetición de un conjunto de Sentencias denominado bloque o *cuerpo del bucle*, mediante la evaluación de una condición o mediante un contador.

Sentencias
WHILE, DO-WHILE y FOR



Sentencia de control repetitiva : FOR

28

Se utiliza para ejecutar un bloque de **Sentencias un número fijo de veces** que se conoce de antemano.

sentencia for

```
for ( inicialización ; condición ; actualización ){  
    Sentencia 1;  
    ...  
    Sentencia n;  
}
```



Sentencia de control repetitiva : FOR

29

La Sentencia **for** se vale de una **variable de control del ciclo**. El ciclo se repite desde el límite inferior, hasta que la variable de control llegue la límite superior.

```
for ( inicialización; condición; actualización ) {  
    Sentencia 1;  
    ...  
    Sentencia n;  
}
```

Cabecera del **for**



Sentencia de control repetitiva : FOR

30

En la cabecera del **for** nos podemos encontrar con 3 partes o secciones

for (inicialización; condición; actualización)

- **Parte de inicialización:** Inicializa la variable de control del bucle. Se pueden utilizar una o varias variables.
- **Parte de condición:** Expresión lógica. El cuerpo del bucle se repite mientras la expresión sea verdadera.
- **Parte de actualización:** Incrementa o decrementa el valor de la variable o variables de control.

```
int main(){
    ....
    for (int i = 0; i<10; i++) {
        cout << "Número: " << i ;
        cout << endl;
    }
    ....
    return 0;
}
```



¿Cuál será la salida?

31

□ int main() {

```
    for(int i=0; i<10; i++) {  
        cout << "Dentro del bucle, i = " << i << endl;  
    }  
    cout << i << endl;  
}
```

ERROR

□ int main() {

```
    for(int i=0; i<10; i++) {  
        cout << "Dentro del bucle, i = " << i << endl;  
    }  
    cout << "ADIÓS" << endl;  
}
```



Sentencia de control repetitiva: FOR

32

```
int main(){
    ....
    for (int n = 1; n<=10; n=n+2)
    {
        cout << "Número: " << n <<"\t" << n*n ;
        cout <<< endl;
    }
    ....
    return 0;
}
```

La variable de control es n

Es de tipo entero, su valor inicial es 1

Su valor final es 10

Se incrementa de 2 en 2.

Pasada	Valor de n
1	1
2	3
3	5
4	7
5	9

En la primera iteración ó pasada, el valor de n es 1, en la segunda pasada n vale 3. La última pasada se realiza cuando n vale 9.



Sentencia de control repetitiva : FOR

33

Se puede inicializar más de una variable, se puede poner más de una condición ó más de un incremento.

```
int main(){
    ....
    for ( int i = 0, j = 100; i<j ; i++, j--) {
        int k;
        k = i+2*j ;
        cout << k << endl;
    }
    ....
    return 0;
}
```

Se declaran 2 variables de control: *i* y *j*

Se inicializan a 0 y 100
respectivamente

El cuerpo se ejecutará
mientras *i* sea menor que *j*

Se incrementa la variable *i* y se
decrementa la variable *j*



Sentencia de control repetitiva : FOR

34

```
int main(){
    ....
    for ( int i = 0, j = 5; i+j <100 ; i++, j =j+5 ) {
        cout << i <<"\t" <<j << (i+j) ;
        cout << endl;
    }
    ....
    return 0;
}
```

Valor de las variables
de control en cada pasada

valor de <i>i</i>	valor de <i>j</i>	<i>i+j</i>
0	5	5
1	10	11
2	15	17
3	20	23
4	25	29
5	30	35
6	35	41
7	40	47
8	45	53
9	50	59
10	55	65
11	60	71
12	65	77
13	70	83
14	75	89
15	80	95

Se realizan un total
de 16 pasadas.

Se ejecuta el cuerpo de la Sentencia **for** mientras
se cumpla la condición.



Sentencia de control repetitiva : FOR

35

Se puede omitir cualquiera de las partes de la cabecera

```
int main(){
    ....
    contador = 1;
    for ( ; contador < 10 ; ) {
        cout << contador ;
        contador++ ;
    }
    ....
    return 0;
}
```

```
int main(){
    ....
    for ( ; ; ) {
        ...
    }
    ....
    return 0;
}
```

Bucle infinito

Bucle infinito que espera la pulsación de una tecla

```
□ for( ; ; ) {
    if(getch() != 0) break;
}
```



Ejemplo

36

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

int main() {
    int numero, acumulador=0, contador=0;
    cin>>numero;
    for(int i=numero; i>0; i--) {
        acumulador+= numero;
        contador++;
        numero--;
    }
    cout<<acumulador;
    system("PAUSE");
    return 0;
}
```



¿Cuál será la salida?

37

```
for(int i=1;i<= impar ;i=i+2) {  
    cout<<setw(MARGEN)<<" ";  
    for(int j=1; j<=i; j++)  
        cout<<"*";  
    cout<<endl;  
  
}  
for(int i=impar-2;i>=1 ;i=i-2) {  
    cout<<setw(MARGEN)<<" ";  
    for(int j=1; j<=i; j++)  
        cout<<"*";  
    cout<<endl;  
}
```



Ejercicios

38

- Múltiplos de 3 menores o iguales a 20
- Leer 50 caracteres del input y hacer un resumen de las letras minúsculas, mayúsculas, dígitos y otros caracteres
- N primeros términos de la sucesión de Fibonacci
- Potencia, x^n
- Factorial de un número $n! = n*(n-1)*(n-2)*(n-3)* \dots 3*2*1$
- Escribe un programa que calcule la suma de todos los números entre 1 y 100 que son múltiplos de 5 o de 7.
- Escribir un programa que calcule el sumatorio $\sum_{i=0}^{10} i^3$
- Tablas de multiplicar



Hacer un programa que genere y escriba lo siguiente

39

1	1	
22	12	9
333	123	8
4444	1234	7
55555	12345	6
666666	123456	5
7777777	1234567	4
88888888	12345678	3
999999999	123456789	2
		1

.- Escribir un programa que dados dos números enteros positivos m y n escriba en pantalla una tabla con los primeros números naturales escritos de forma correlativa.

Por ejemplo m=3 y n=4 (m número de filas y n el número de columnas)

1	2	3	4
5	6	7	8
9	10	11	12



Sentencia de control repetitiva : WHILE

40

estructura algorítmica *hacer_mientras*:

sentencia while

```
while ( condición ) {  
    Sentencia 1;  
    ...  
    Sentencia n;  
}
```

```
while ( condición )  
    Sentencia;
```

Cada repetición del cuerpo del bucle se **denomina iteración**

Las Sentencias se ejecutan mientras se cumpla la **condición**, que será evaluada siempre **antes** de cada repetición.



Sentencia de control repetitiva : WHILE

41

```
#include <iostream>
using namespace std;
int main(){
    ...
    contador = 0;
    while (contador < 100) {
        cout << "Hola";
        contador ++;
    }
    ...

    return 0;
}
```

La condición será evaluada siempre antes de cada iteración.

El cuerpo del bucle while se repite mientras la condición sea cierta.

El cuerpo de un bucle **while** se ejecutará **cero o más veces**.



Sentencia de control repetitiva : WHILE

42

Inicialización

Se realiza antes de la Sentencia **while**

```
...  
contador = 0;  
while (contador < 100) {  
    cout << "Hola";  
    contador ++;  
}  
...
```

cuerpo

La variable que representa la condición del bucle se denomina **variable de control del bucle**.

Dicha variable debe ser:

- inicializada
- comprobada
- actualizada

Actualización

Se realiza dentro del cuerpo del bucle durante cada iteración

Condición

Se comprueba el valor de la variable antes de comenzar la repetición



Sentencia de control repetitiva : WHILE

43

```
int main(){
    const bool continuar = true;
    bool a = true;
    int n;
    while (a ==continuar) {
        cin >> n;
        if (n<0) {
            cout << "No se admiten negativos";
            a = false;
        }
        else
            cout << "Uno más: " ;
    }
    return 0;
}
```

Inicialización

Comprobación

Es importante comprobar que en algún momento, la condición del bucle se hace falsa, de forma que no obtengamos bucles infinitos.

Actualización



C++

Escribir Hola cont-veces

44

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(){
    int cont;
    cout<<"Numero de veces a repetir hola"<<endl;
    cin>>cont;
    while (cont >0){
        cout<<"Hola ";
        cont=cont-1;
    }
    system("pause");
    return 0;
}
```



Raíz cuadrada de un número dado

45

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
    int raiz=0,num;
    cin>>num;
    while( (raiz+1)*(raiz+1) <= num)
        raiz ++;
    cout<<raiz;
    system("pause");
    return 0;
}
```



Calcular la media de n-números

46

```
#include <iostream>
using namespace std;
int main()
{
    float num, media,suma=0;
    int n, cont=0;
    cout<<" Cantidad de números a sumar "<<endl;
    cin>>n;
    while (cont<n) {
        cout<<"Dime un numero "<<endl;
        cin>>num;
        suma=suma+num;
        cont=cont+1;
    }
    media=suma/n;
    cout<<"MEDIA= "<<media<<endl;
    system("pause");
    return 0;
}
```



Ejemplos while

47

- ❑ Divisores de un número
- ❑ Comprobar si un número dado es primo
- ❑ Número de cifras de un número
- ❑ Encontrar el mayor de una serie de números dados por teclado
- ❑ Dado un entero positivo, escribir todos los números enteros positivos desde el número dado al 0.
- ❑ Máximo común divisor de dos números
 - ▣ Algoritmo de Euclides
 - ▣ Otro método
- ❑ Mínimo común múltiplo
$$\text{mcm}(m,n) * \text{mcd}(m,n) = m*n$$



Sentencia de control repetitiva : DO-WHILE

48

sentencia do-while

```
do{  
    Sentencia 1;  
    ...  
    Sentencia n;  
} while ( condición );
```

```
do  
    Sentencia;  
while ( condición );
```

- Ejecuta un bloque de Sentencias al menos una vez.
- El cuerpo del bucle **se repite mientras se verifica la condición**.
- La condición será evaluada **después** de cada repetición.



Sentencia de control repetitiva : DO-WHILE

49

```
#include <iostream>
int main()
{
    char n;
    do {
        cout << "introducir número: ";
        cin >> n;
        if ((isdigit(n)) == false)
            cout << "Solo se admiten números";
    } while ((isdigit(n)) != false);

    return 0;
}
```

cuerpo

El cuerpo de un bucle **do-while** se ejecutará una o más veces.

El cuerpo del bucle **do-while** se repite mientras la condición sea cierta.

La condición será evaluada siempre después de cada iteración.



Leer un número entero en un intervalo

50

```
#include <iostream>
using namespace std;
int main() {
    int num, min=1, max=100;
    do {
        cout<<"un número entre 1 y 100: ";
        cin>>num;
    } while(num<1 || num>100);

    system("pause");
    return 0;
}
```



Ejemplo do... while

51

- ❑ Escribir los múltiplos de un número dado
- ❑ Leer un entero positivo
- ❑ Solicitar números al usuario y escribir el primero mayor que 100
- ❑ Solicitar números al usuario y sumar los impares, la entrada terminará cuando escriba el cero
- ❑ Mostrar un menú al usuario, que seleccione una opción y la ejecute, el programa se ejecutará hasta que el usuario seleccione el fin.



Diferencias entre las Sentencias WHILE y DO-WHILE

52

En **while** primero se tiene que cumplir la condición, y luego se ejecuta el bloque de Sentencias.

```
while ( condición ) {  
    Sentencia 1;  
    ...  
    Sentencia n;  
}
```

```
if (condición)  
    do {  
        Sentencia 1;  
        ...  
        Sentencia n;  
    } while ( condición );
```

En **do-while** primero se ejecuta el bloque de Sentencias y luego se comprueba la condición.



Errores comunes en los bucles

53

Cuerpo del bucle vacío

```
while (num !=0);  
{ cout<<"hola"<<num;  
  cin>>num;  
}  
  
for (i=0 ; i <=10; i++);  
{ cubo = i * i * i;  
  suma = suma + cubo;  
}
```

Bucles que nunca acaban

```
for (i=0 ; i >=0; i++)  
{ cubo = i * i * i;  
  suma = suma + cubo;  
}
```

Bucles donde nunca se ejecuta el cuerpo del bucle.

```
for (i=0 ; i >= 10; i++){  
  cubo = i * i * i;  
  suma = suma + cubo;  
}
```

Una iteración de más o una iteración de menos

p.e. cálculo de la potencia



Controlar el fin de línea

54

- ❑ `c=cin.peek();` librería `iostream`
- ❑ Devuelve el carácter leído o EOF en caso de haber alcanzado el fin de archivo.
- ❑ **Nota:** No retira el carácter del flujo de entrada.

`while((cin.peek())!= '\n')`

También podríamos hacerlo con:

`while (cin>>num)` al final de la entrada de datos
tenemos que dar CTRL+Z



Ejercicio

55

Una empresa textil tiene una serie de datos que indican si el producto al cual hace referencia pasa los controles de calidad o no. Estos datos vienen dados por los caracteres 'A' o 'B' de manera que el indicador 'A' significa que el producto al cual hace referencia pasa los controles de calidad, pero el indicador 'B' significa que el producto no pasa los controles de calidad. Suponiendo que el último dato (el centinela) es el carácter 'C', diseña un programa que indique el porcentaje de productos que pasan los controles de calidad.



```
#include <iostream>
int main(){
    char caracter;
    int totalA = 0, totalB = 0, proporcion=0;
    cout << "Introcuzca los controles de los productos: ";
    cin >> caracter;
    while ((caracter != 'A') && (caracter != 'B') && (caracter != 'C')) {
        cin >> caracter;
    }
    while (caracter != 'C') {
        if (caracter == 'A')
            ++totalA;
        else ++totalB;
        cin >> caracter;
        while ((caracter != 'A') && (caracter != 'B') && (caracter != 'C')) {
            cin >> caracter;
        }
    }
    proporcion = (totalA*100)/(totalA+totalB);
    cout << "El porcentaje de de productos que pasan los controres de calidad es " << proporcion <<
    "%." << endl;
    return 0;
}
```



Sumatorio de la página 38

57

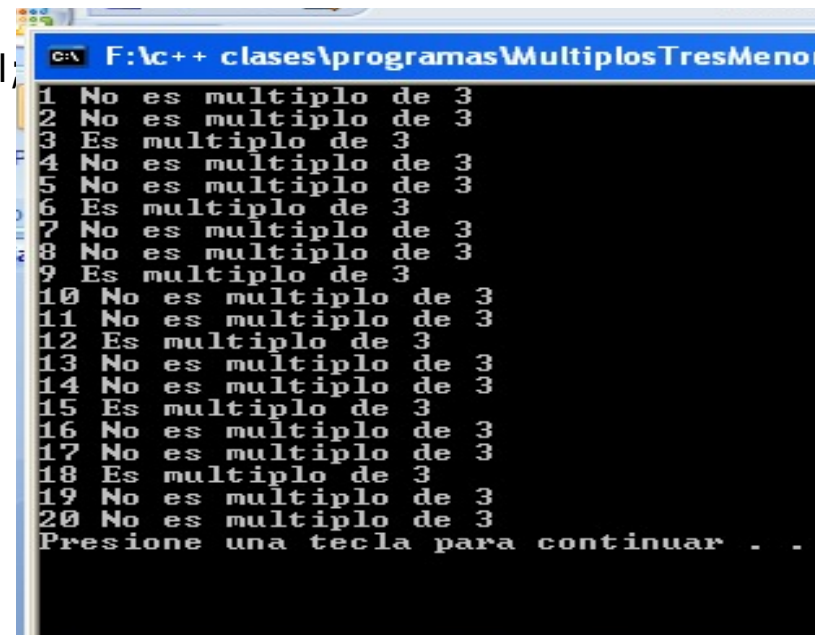
```
int main()
{
    int i, cubo, suma=0;
    for (i=0 ; i <=10; i++) {
        cubo = i * i * i;
        suma = suma + cubo;
    }
    Cout<<"el sumatorio es", suma);
}
```



Multiplos de 3 menores o iguales a 20

58

```
#include <iostream>
using namespace std;
int main ()
{
    for (int i=1;i<=20;i++)
    {
        cout<<i;
        if (i%3==0) cout<<" Es multiplo de 3" <<endl;
        else cout<<" No es multiplo de 3"<<endl;
    }
    system("pause");
    return 0;
}
```



```
C:\ F:\c++ clases\programas\Multiplos Tres Meno
1 No es multiplo de 3
2 No es multiplo de 3
3 Es multiplo de 3
4 No es multiplo de 3
5 No es multiplo de 3
6 Es multiplo de 3
7 No es multiplo de 3
8 No es multiplo de 3
9 Es multiplo de 3
10 No es multiplo de 3
11 No es multiplo de 3
12 Es multiplo de 3
13 No es multiplo de 3
14 No es multiplo de 3
15 Es multiplo de 3
16 No es multiplo de 3
17 No es multiplo de 3
18 Es multiplo de 3
19 No es multiplo de 3
20 No es multiplo de 3
Presione una tecla para continuar . .
```



Al inventor del juego del ajedrez le dijeron que pidiera lo que quisiera, que estaba concedido y se le ocurrió pedir: 1 gramo de trigo por la primera casilla, 2 gramos por la segunda, 4 por la tercera, 8 por la cuarta, 16 por la quinta y así sucesivamente. Se pide un programa que escriba una matriz de ocho por ocho enteros, representando en cada posición la cantidad de granos correspondientes a la casilla respectiva:

1	2	4	8	16	32	64	
128		..	..	..	...	..	..
..	..	..	...	..	..	..	

¿Se puede hacer alguna observación sobre este problema?



- Una **terna pitagórica** es un conjunto de tres números enteros positivos tales que .

Encontrar ternas pitagóricas hasta el 100.

$$x^2 + y^2 = z^2$$



Tabla de multiplicar

61

```
//Tablas de multiplicar
#include <iostream>
using namespace std;
int main()
{
    for(int i=1; i<=10; i++)
    {
        for(int j=1; j<=10; j++)
            cout<<i<<"x"<<j<<"="<<i*j<<endl;
        system("pause");
    }
    system("pause");
    return 0;
}
```



Factorial

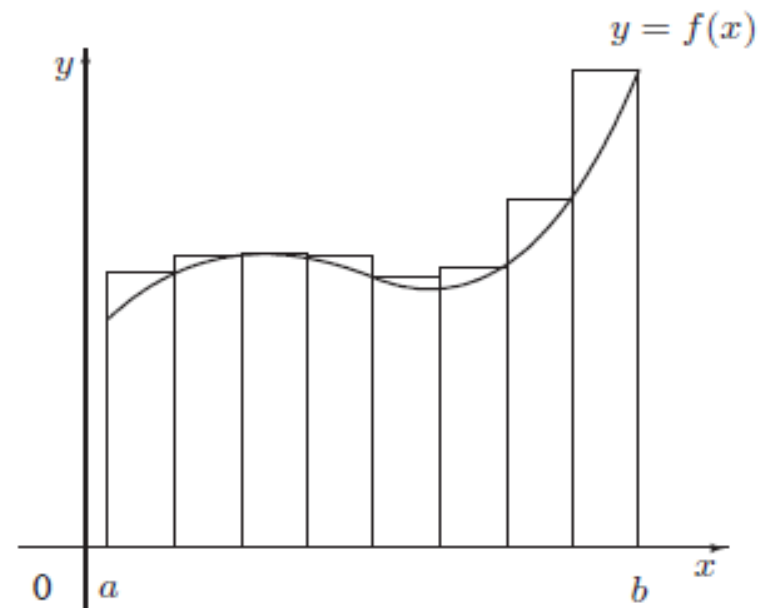
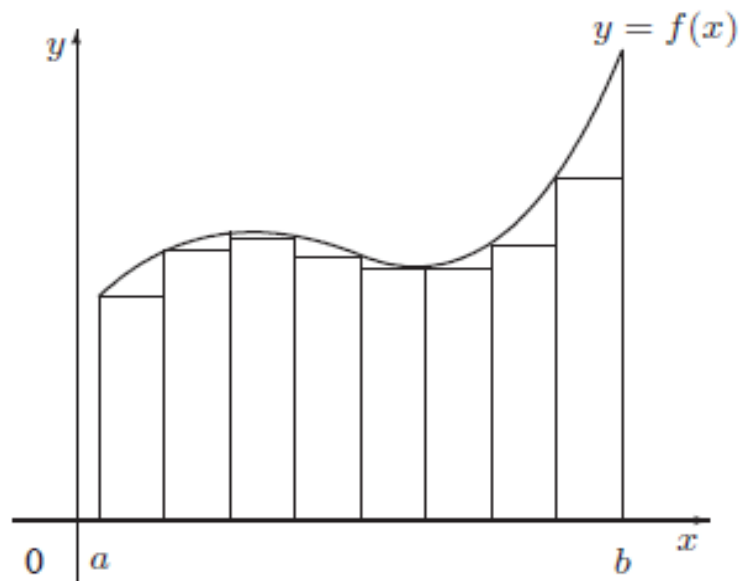
62

```
//Factorial de un numero
#include <iostream>
using namespace std;
int main()
{
    int num,factorial=1;
    cout<<"Dame un número"<<endl;
    cin>>num;
    for(int i=2; i<=num; i++)
        factorial=factorial*i;
    cout<< num<<"!= "<<factorial<<"\n";
    system("pause");
    return 0;
}
```



Integral

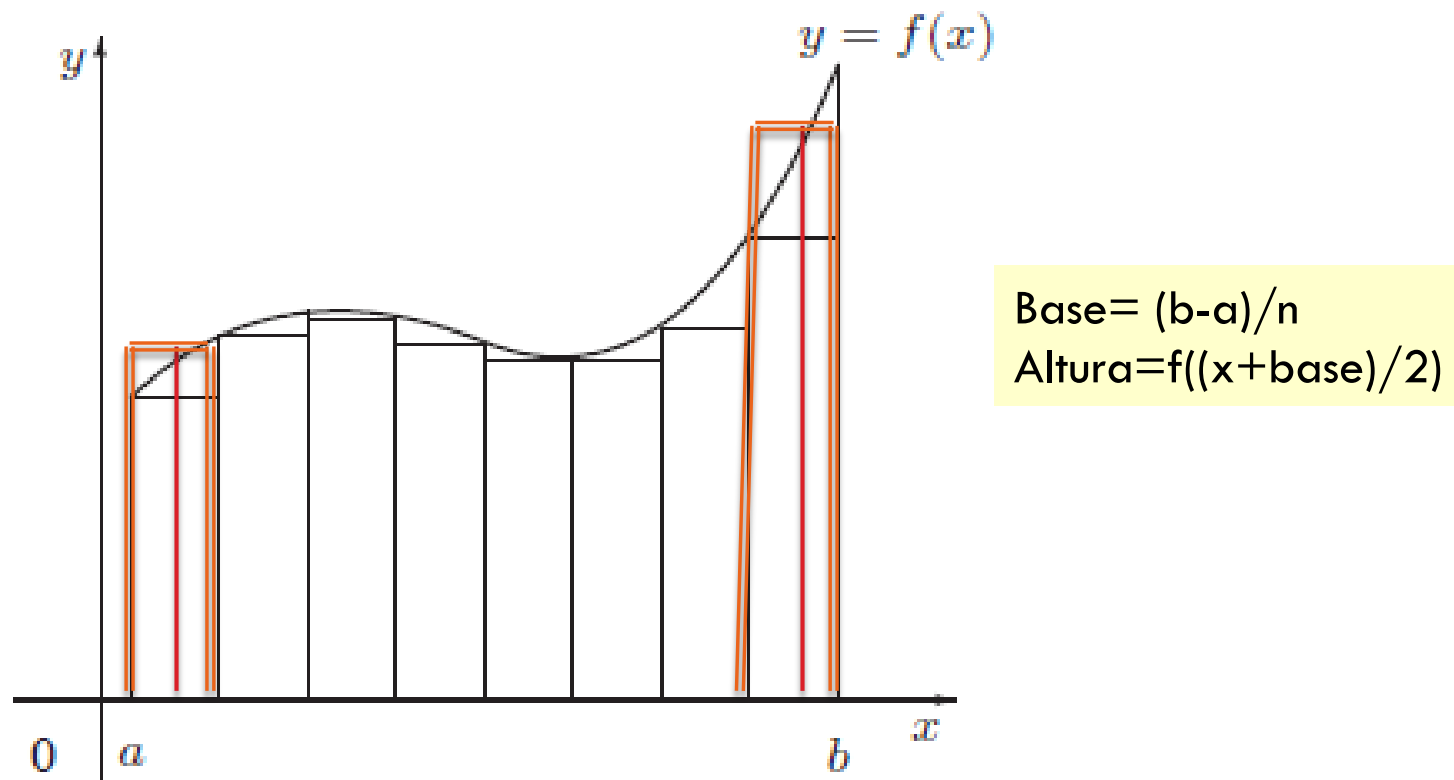
63



C++

Integral

64



Integral = suma de las áreas de los rectángulos



C++

Integral

65

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
    float a,b, base, altura, x, area=0;
    int n;
    cout<<"Dame el extremo inferior del intervalo, a= ";
    cin>>a;
    cout<<"Dame el extremo superior del intervalo, b= ";
    cin>>b;
    cout<<"Número de rectángulos a dividir ";
    cin>>n;
    base=(b-a)/n;
    x=a;
    for (int i=1; i<=n; i++){
        altura=(x+base/2)*(x+base/2);
        area=area+ base* altura;
        x=x+base;
    }
    cout<<area<<endl;

    system ("PAUSE");
    return 0;
}
```



Potencia

66

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
    double potencia=1, b, base;
    int exponente,n;
    cout<<"base";
    cin>>base;
    cout<<"base";
    cin>>exponente;
    n=exponente;
    b=base;
    if(n<0) {
        b=1.0/b;
        n=-n;
    }
    for (int i=1;i<=n;i++){
        potencia=potencia*b;
    }
    cout<<base<<"^"<<exponente<<"= "<<potencia<<endl;
    system ("PAUSE");
    return 0;
}
```

