

26

## 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**



C++

# 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;  
}
```



**C++**

# 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;
}
```



C++

## ¿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;  
}
```



**C++**

# 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*



C++

# 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.



C++

# 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;
}
```



C++

# 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;
}
```



C++

## ¿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;  
}
```



C++

# 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

C++

## Hacer un programa que genere y escriba lo siguiente

39

```
1          1
22         12
333        123
4444       1234
55555      12345
666666     123456
7777777    1234567
88888888   12345678
999999999  123456789

          9
          8
          7
          6
          5
          4
          3
          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
```



C++

# 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.



C++

# 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**.



C++

# Sentencia de control repetitiva : WHILE

42

## Inicialización

Se realiza antes de la Sentencia **while**

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

cuerpo

## Actualización

Se realiza dentro del cuerpo del bucle durante cada iteración

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

Dicha variable debe ser:

- inicializada
- comprobada
- actualizada

## 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;
}
```



C++

# 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;
}
```



C++

# 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;
}
```



C++

# 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
$$\text{mcm}(m,n) * \text{mcd}(m,n) = m*n$$



C++

# 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.



C++

# 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.



C++

## 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;
}
```



C++

## 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.



C++

## 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



C++

## 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.



C++

```
#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);
}
```

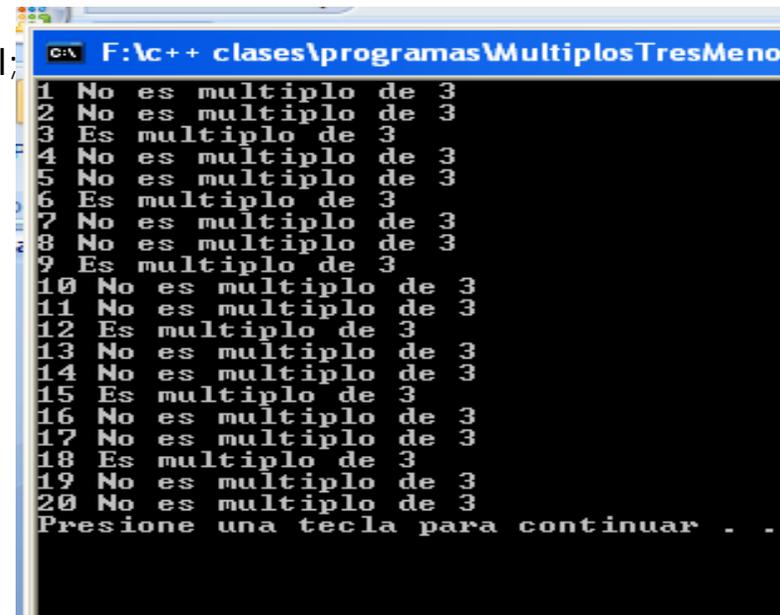


C++

# 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;
}
```



```
F:\c++ clases\programas\MultiplosTresMeno
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;
}
```



C++

# Factorial

62

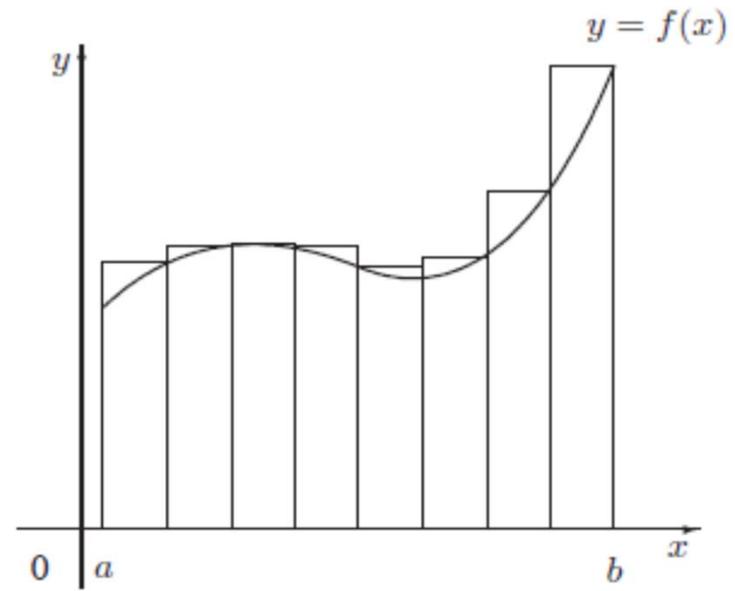
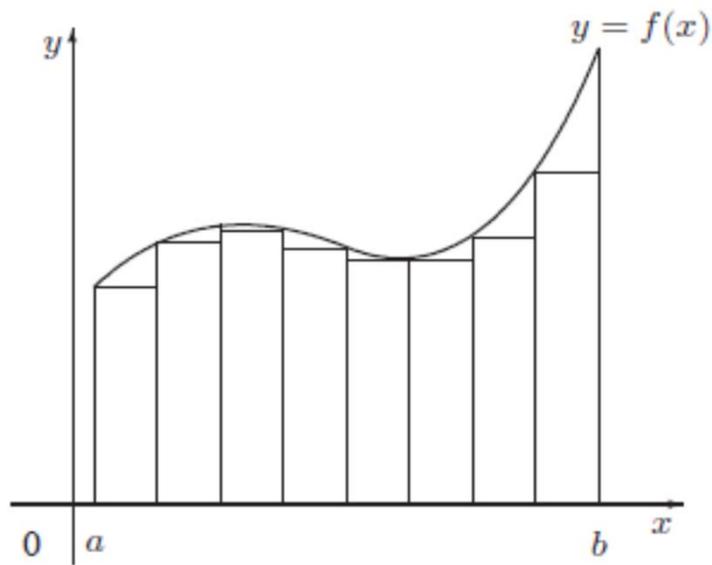
```
//Factorial de un numero
#include <iostream>
using namespace std;
int main()
{
    int num,factorial=1;
    cout<<"Dame un número"<<endl;
    cin>>num;
    for(int i=2; i<=num; i++)
        factorial=factorial*i;
    cout<< num<<"! = "<<factorial<<"\n";
    system("pause");
    return 0;
}
```



C++

# Integral

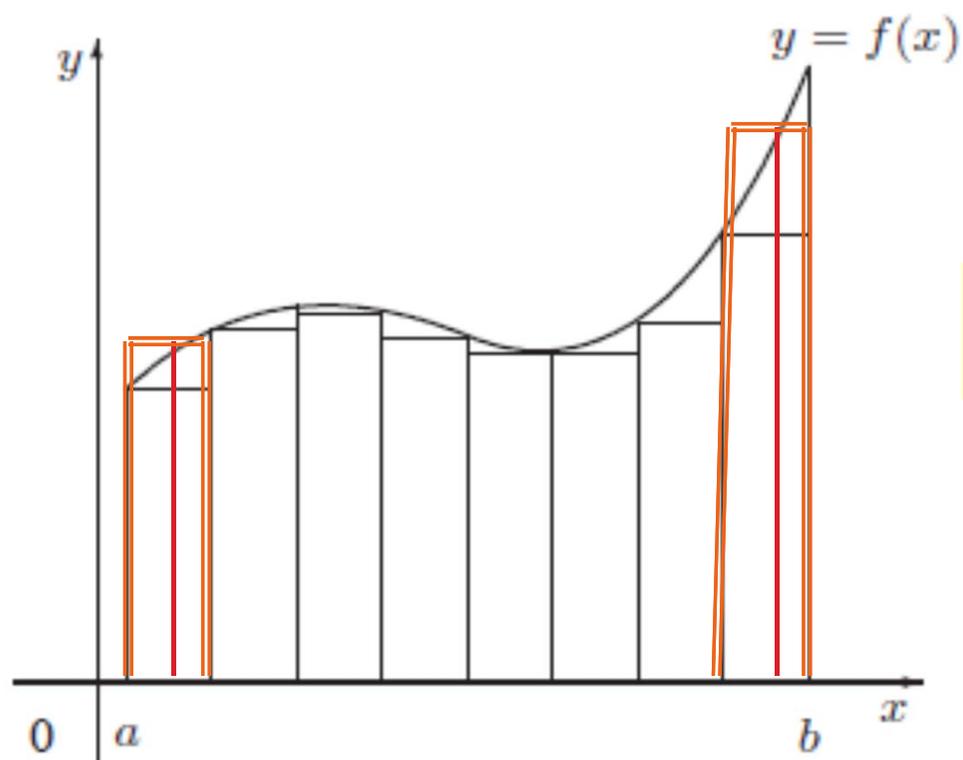
63



C++

# Integral

64



$$\text{Base} = (b-a)/n$$
$$\text{Altura} = f((x+\text{base})/2)$$

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;
}
```



C++

# 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;
}
```



C++