

al-go-rithm (al'ge-rith'm)

n.

A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps.

Ejemplo: algoritmo para multiplicar dos números y presentar el resultado en pantalla:

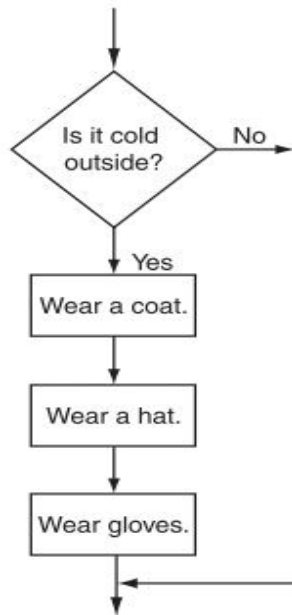
- 1.- Presentar un texto diciendo lo que el programa va a hacer.
- 2.- Pedir el primer número.
- 3.- Almacenar el primer número.
- 4.- Pedir el segundo número.
- 5.- Almacenar el segundo número.
- 6.- Multiplicar los números y almacenar el resultado.
- 7.- Presentar el resultado en la pantalla.

Ejemplos de algoritmos famosos:

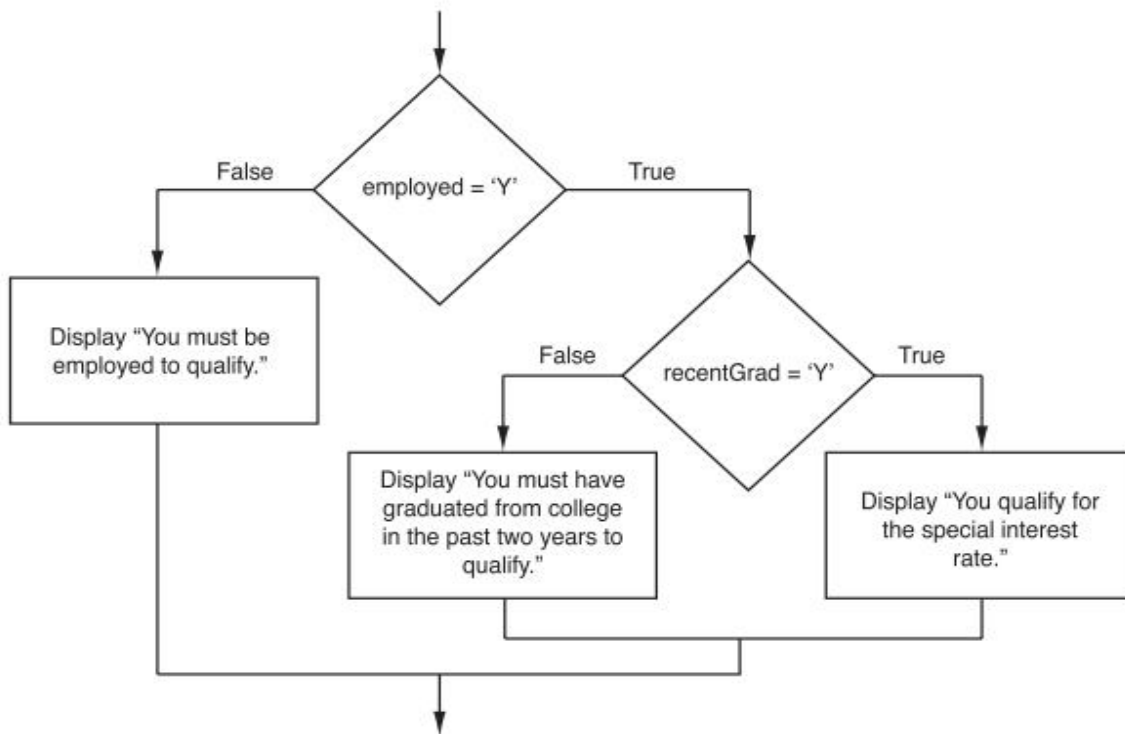
- the algorithm does not require factoring the two integers.
12. **Expectation-maximization algorithm (EM-Training)**
In statistical computing, an expectation-maximization (EM) algorithm is an algorithm for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. EM alternates between performing an expectation step, which computes the expected value of the latent variables, and a maximization step, which computes the maximum likelihood estimates of the parameters given the data and setting the latent variables to their expectation.
 13. **Fast Fourier transform (FFT)**
Efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. FFTs are of great importance to a wide variety of applications, from digital signal processing to solving partial differential equations to algorithms for quickly multiplying large integers.
 14. **Gradient descent**
Gradient descent is an optimization algorithm that approaches a local minimum of a function by taking steps proportional to the negative of the gradient (or the approximate gradient) of the function at the current point. If instead one takes steps proportional to the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.


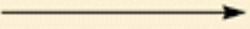

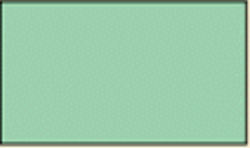
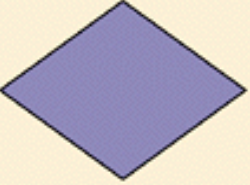
Para facilitar el uso de algoritmos y crear código de programación eficiente, se usan los diagramas de Flujo.

Ejemplo:



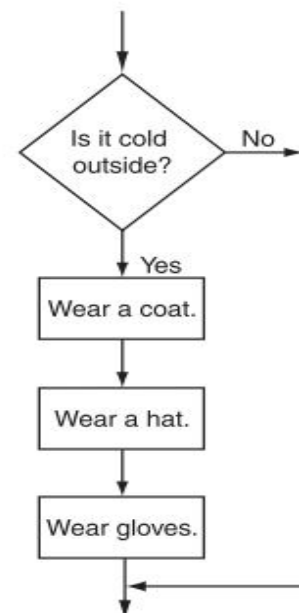
Otro ejemplo



Name	Symbol	Use in flowchart
Oval		Denotes the beginning or end of a program.
Flow line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation (e.g., INPUT) or an output operation (e.g, PRINT).
Rectangle		Denotes a process to be carried out (e.g., an addition).
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).

Ejercicio: crear el algoritmo que representa el siguiente diagrama de flujo e implementarlo en C++:

- preguntar hace frio fuera?.
- en caso afirmativo.
- imprimir que lleve un abrigo.
- imprimir que lleve un sombrero.
- imprimir que lleve guantes.
- darle autorización para salir.
- en caso negativo.
- darle autorización para salir.



```

1 // enteros y muestra el resultado en pantalla
2 // enteros y muestra el resultado en pantalla
3 #include <iostream>
4 using namespace std;
5 int main() {
6     int frio;
7     cout << "Hace frio afuera, entre 1 si es correcto o 0 si es incorrecto\n";
8     cin >> frio;
9     if(frio == 1){
10        cout << "Pongase su abrigo por favor.\n";
11        cout << "Pongase su sombrero por favor.\n";
12        cout << "Pongase sus guantes por favor.\n";
13    }else{
14        cout << "No se ponga nada, no hace frio.\n";
15    }
16    cout << "Tiene usted autorización para salir.\n";
17    return 0;
18 }

```

```

if (condition)
{
    statement set 1;
}
else
{
    statement set 2;
}

```

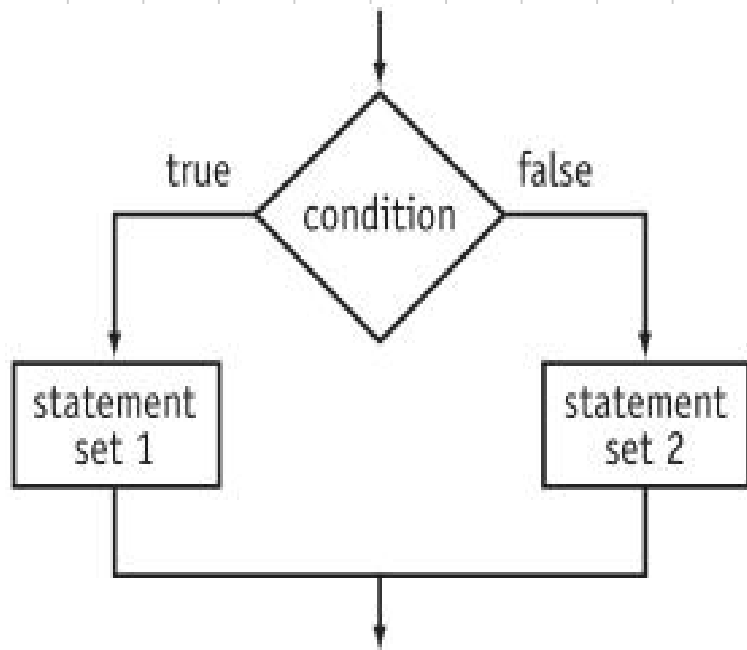
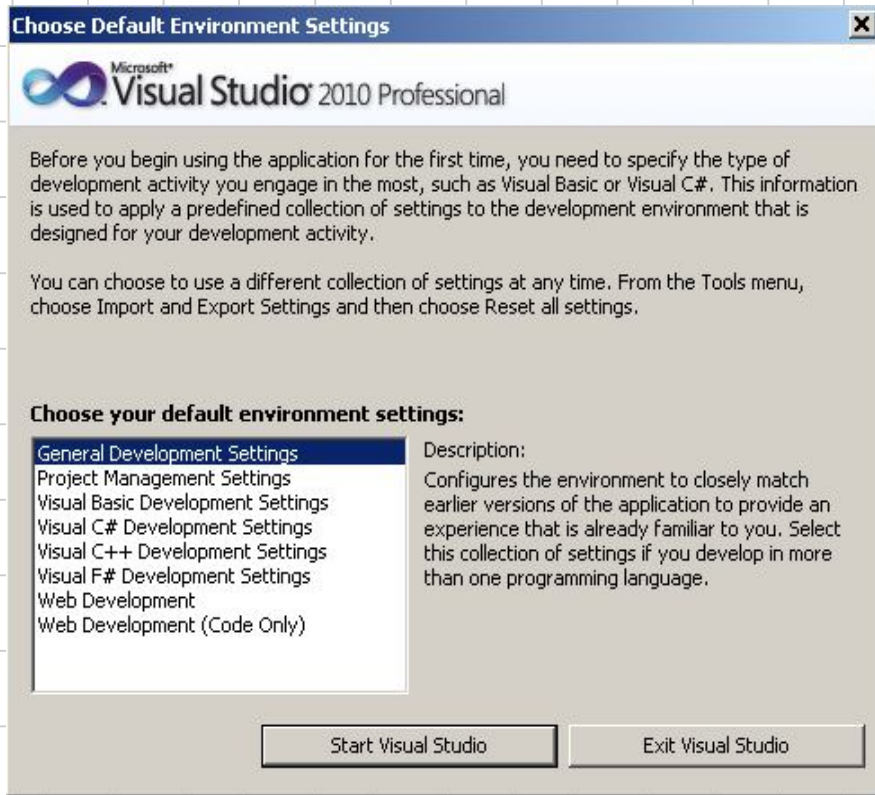


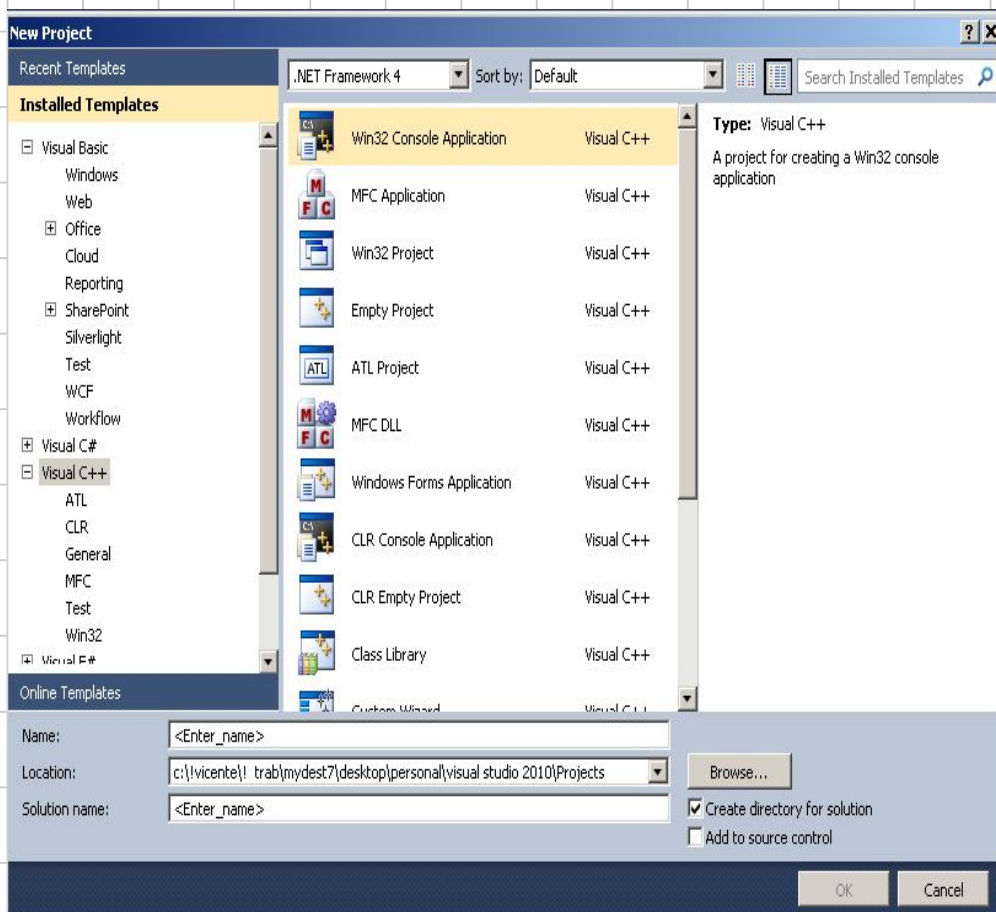
Table 2-2 Common Escape Sequences

Escape Sequence	Name	Description
\n	Newline	Causes the cursor to go to the next line for subsequent printing.
\t	Horizontal tab	Causes the cursor to skip over to the next tab stop.
\a	Alarm	Causes the computer to beep.
\b	Backspace	Causes the cursor to back up, or move left one position.
\r	Return	Causes the cursor to go to the beginning of the current line, not the next line.
\\	Backslash	Causes a backslash to be printed.
\'	Single quote	Causes a single quotation mark to be printed.
\"	Double quote	Causes a double quotation mark to be printed.

Como abrir el Visual C++. Double click en el Icon de Visual Studio y aparece

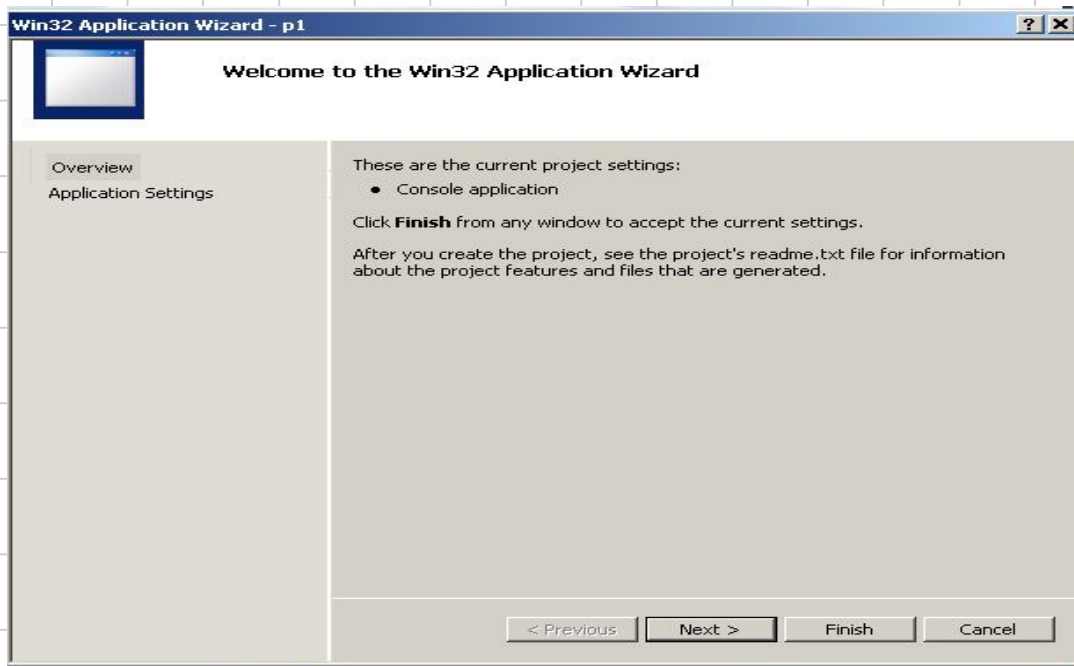


Hay que crearse un nuevo proyecto: en el Menu > File > New y aparece

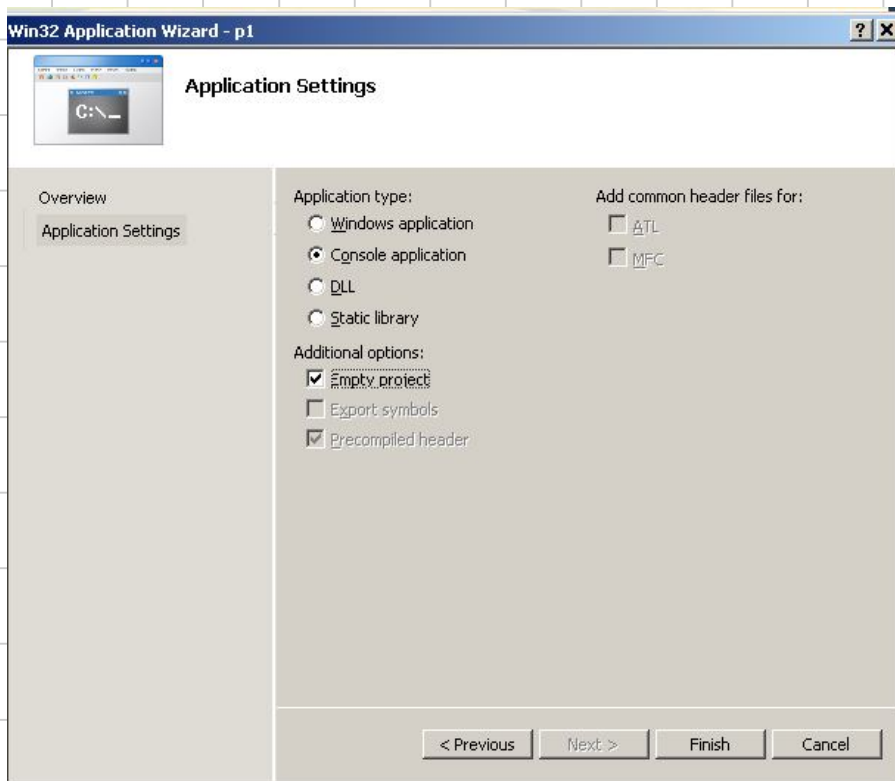


Seleccionen a la izquierda Visual C++ y a la derecha Win32 Console application. Le dan un nombre al proyecto como p1 y le dan al OK

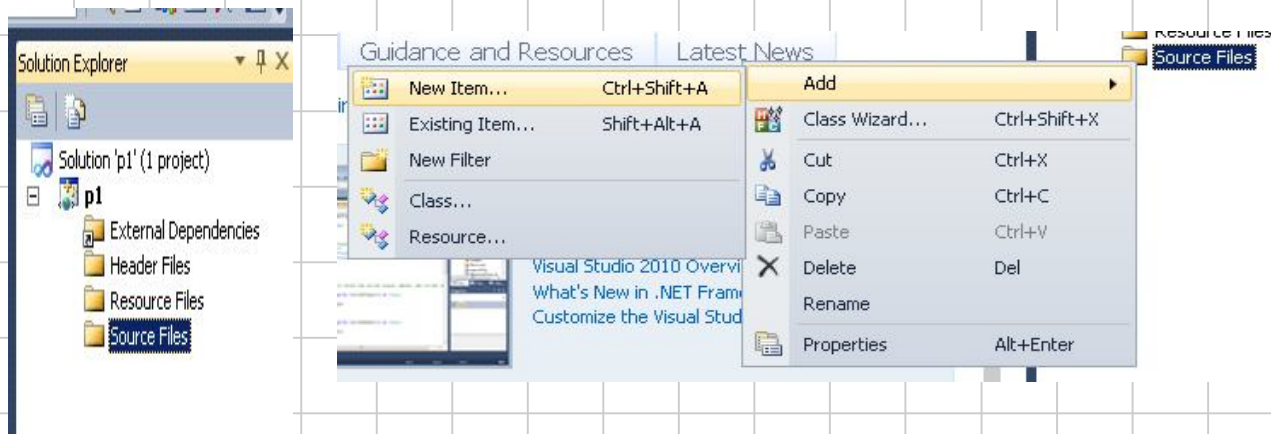
Aparece la siguiente ventana:



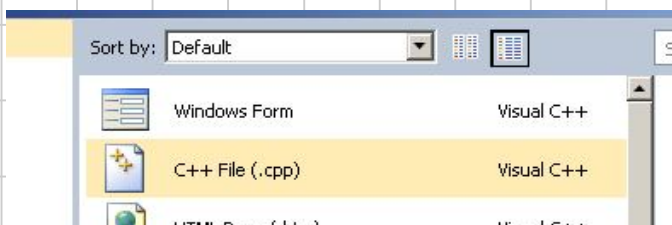
Dan a Next y aparece la siguiente ventana. Seleccione Empty Project y dar a OK



Aparece unas ventanas. Y en una parte que pone Solution Explorer selecciona Source File y dar right click!!
Seleccionar New Item



Una vez se abre, seleccionar C++ File (.cpp) y dar un nombre corto: frio y dar OK



Se abre un fichero frio.cpp y ya pueden empezar a escribir el código

```
Start Page  main.cpp* x
(Global Scope)  main()
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int frio;
5      //char myChar;
6      cout << "Hace frio afuera? Entre 1 si si, o 0 si no hace frio: ";
7      cin >> frio;
8      if(frio == 1){
9          cout << "Cuando el grajo vuela bajo hace un frio del ...\\n";
10
11      }else{
12          cout << "Vamos a piñones!!\\n";
13      }//end if
14      cout << "Enter 0 to finish...";
15      cin >> frio;
16      //myChar = cin.get();
17      return 0;
18  }//endmain
```

Si la ventana de Solution explorer no aparece. Hay que resetear la configuración de las ventanas. En el menú click en Window>Reset Window Layout



Si se quiere que aparezcan las líneas de código, hay que dar al menú Tool>Options y en Text Editor seleccionar C/C++>General y en Display habilitar el Line numbers, tal y como aparece en la siguiente figura:

