

Buenas prácticas de programación

Aunque C/C++ uno puede escribir un programa bastante libremente, existen ciertas normas de escritura que se siguen para evitar errores, y sobre todo que quede claro para que las demás personas que lean el programa no se pierdan (en especial los profesores)

Aunque todas las sentencias se pueden escribir en cualquier lugar del programa y pasan a valer desde ahí en adelante, normalmente se estila colocarlos siempre en el mismo lugar y en grupos.

El orden normalmente es el siguiente:

Primero los includes

Segundo los prototipos de las funciones

Tercero las definiciones de tipo (lo veremos en la unidad de arreglos)

Cuarto las definiciones de variables globales

Quinto las funciones (incluyendo main)

Aunque podrían cambiar el orden de algunos como ponerlos prototipos después de las variables globales, lo importante es que estén todos los prototipos juntos.

Indentación:

La indentación sirve para que los que leen el programa fuente puedan entenderlo más fácilmente. La indentación no le interesa al compilador.

La práctica de indentación se trata de desplazar varios espacio a la derecha las sentencias de tal forma de alinearlas a todas de tal forma que de un vistazo se puedan identificar las sentencias de un mismo nivel.

Normalmente las llaves en C/C++ marcan un cambio de nivel, por ejemplo:

```
int main() {
    int x1,x2,x3,x4,x5; //coordenadas x de los aviones
    int y1,y2,y3,y4,y5; //coordenadas y de los aviones

    x1=x2=x3=x4=x5=2; //todos empiezan en x=2
    y1=2; //empiezan en distintas alturas
    y2=6;
    y3=9;
    y4=14;
    y5=19;
    cout<<clrscr; //borramos la pantalla
    for(int n=2;n<70;++n)
    {
        imprBombardero(x1,y1); //imprimimos el avión número 1
        imprBombardero(x2,y2); //imprimimos el avión número 2
        imprBombardero(x3,y3); //imprimimos el avión número 3
        imprBombardero(x4,y4); //imprimimos el avión número 4
        imprBombardero(x5,y5); //imprimimos el avión número 5
    }
}
```

```

        ++x1; //avanzamos avión 1
        ++x2; //avanzamos avión 2
        ++x3; //avanzamos avión 3
        ++x4; //avanzamos avión 4
        ++x5; //avanzamos avión 5
        Sleep(100);
    }

    return 0;
}

```

Pueden verlo en los ejemplos.

Con respecto a la llave que abre las funciones existen dos formas, una es colocarla inmediatamente después del prototipo de la función y otra en la línea siguiente:

```
int main() { //inmediatamente después
```

```
int main() //en la línea siguiente
{
```

Pero el indentado siempre se realiza.

Los casos en que no hay llaves y se indenta es en las sentencias de interacción con una sola sentencia y la sentencia va en la línea siguiente, lo mismo para el if:

```
for(int n=2;n<70;++n)
    cout<<n;
```

```
if(x>10)
    cout<<x;
```

Variables locales

Siempre es conveniente declarar todas las variables locales al principio de la función, se puede hacer una excepción en los for cuando la variable que se usa no se necesita más al final del mismo:

```
for(int n=2;n<70;++n)
    cout<<n;
```