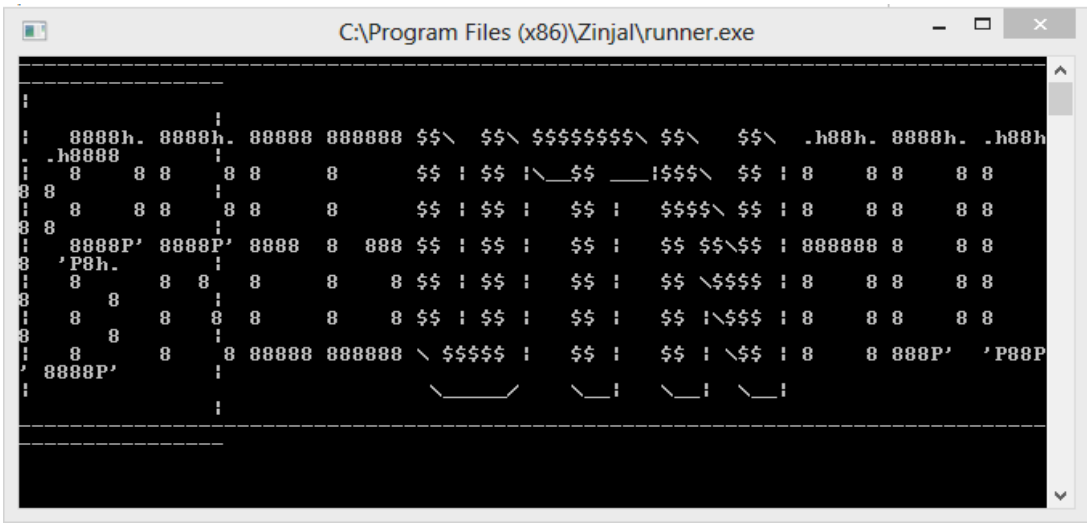
Funciones útiles para TP de Algoritmos y Estructuras de Datos

- *Tamaño consola*

Se puede utilizar la instrucción *system("mode con: cols=ANCHO lines=ALTO");* para modificar las dimensiones de la consola.

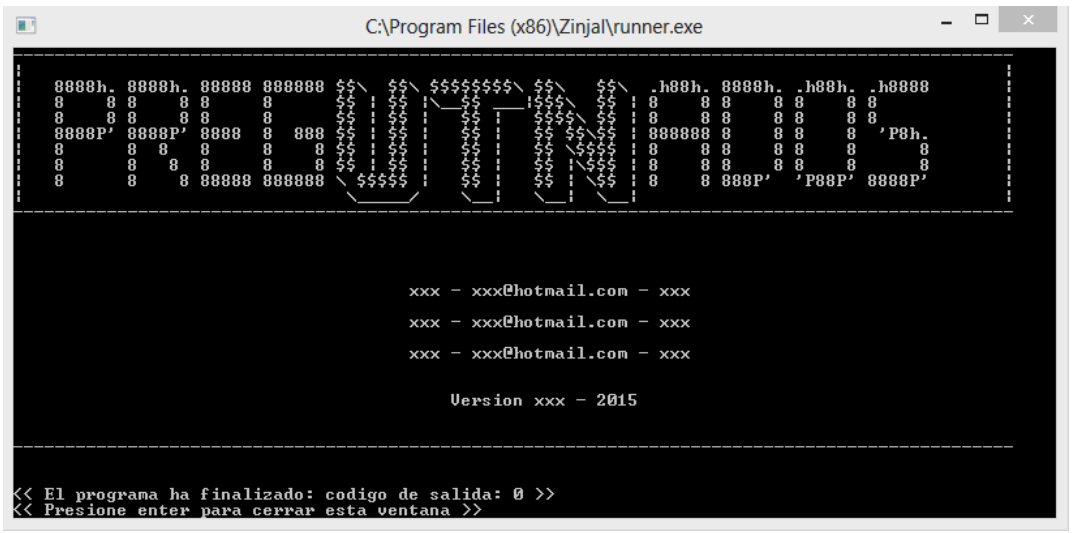
Ejemplo:

Con el tamaño por defecto.



Incluyendo en el inicio de programa la siguiente línea de código:

```
int main(int argc, char *argv[]) {
    system("mode con: cols=100 lines=30");
     logo();
     menu_principal();
    return 0;
}
```



❖ *Cambiar color de letra y fondo de toda la consola*

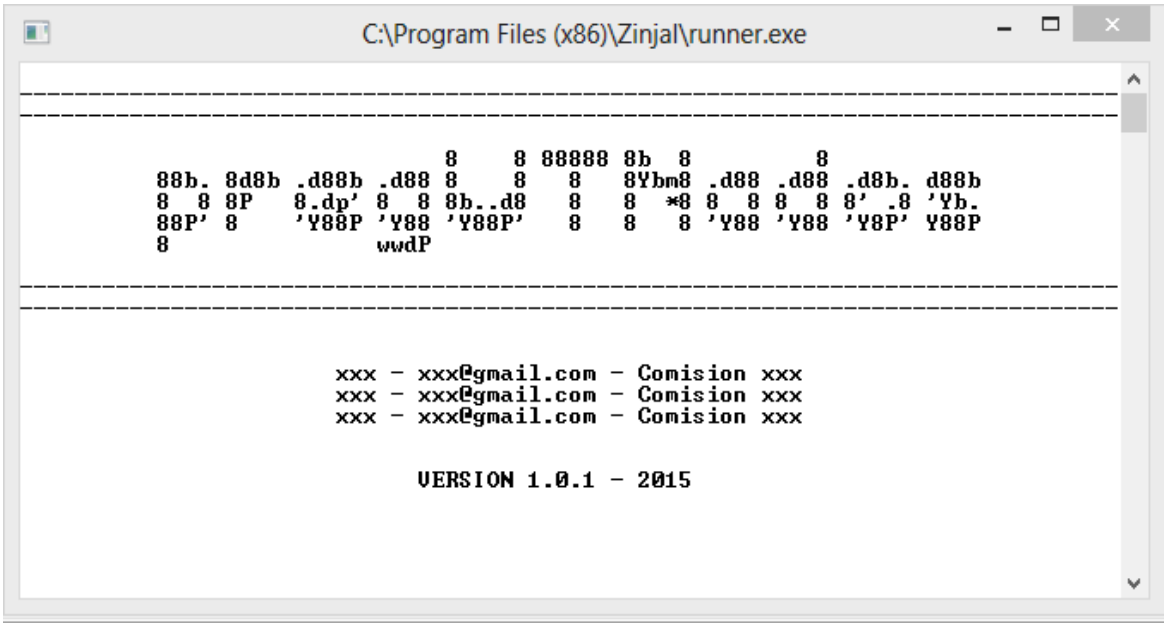
Para cambiar el color del fondo de la consola y del texto pueden utilizar la función system(“color XY”); siendo X un dígito hexadecimal que indica el color del fondo e Y otro dígito hexadecimal que establece el color del texto. Se debe incluir la libreria <cstdlib>.

Colores

0 = Negro	8 = Gris
1 = Azul	9 = Azul claro
2 = Verde	A = Verde claro
3 = Aguamarina	B = Aguamarina claro
4 = Rojo	C = Rojo claro
5 = Púrpura	D = Púrpura claro
6 = Amarillo	E = Amarillo claro
7 = Blanco	F = Blanco brillante

Ejemplo con fondo blanco y letra negra.

```
#include<stdlib.h>
...
int main(int argc, char *argv[]) {
   system("color F0");
    logo();
    menu_principal();
   return 0;
}
```



❖ *Cambiar color de letra y fondo de una parte de la consola*

Si se desea cambiar el color de sólo una parte de la consola se puede utilizar la función:

SetConsoleTextAttribute(GetStdHandle (STD_OUTPUT_HANDLE),X);

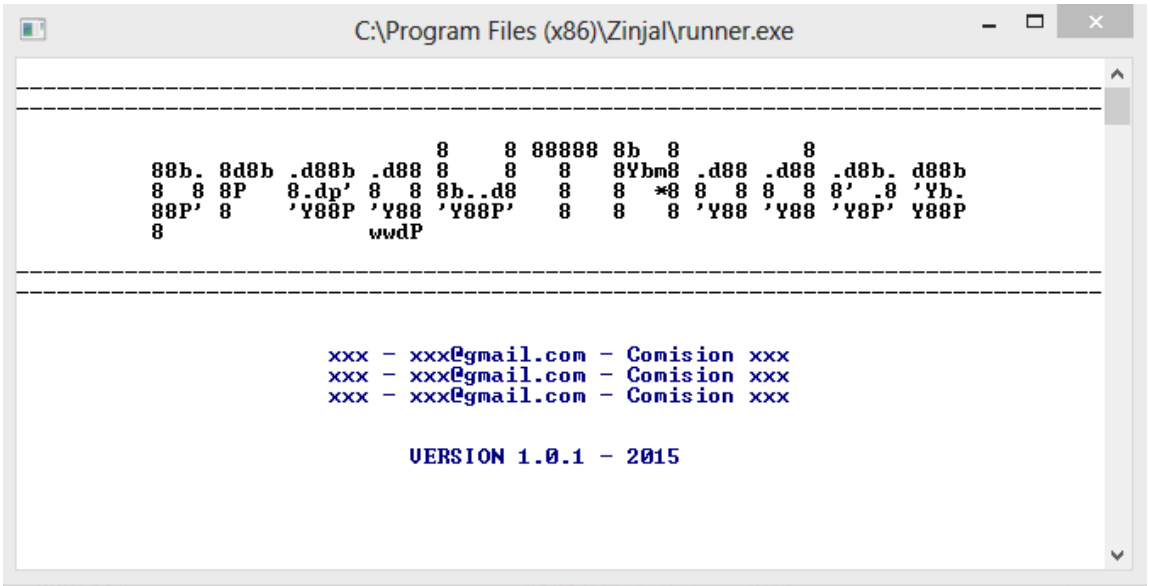
El valor de X debe estar entre 0 y 255 y se calcula de la siguiente manera:

X=W*16+Z;

Siendo W=Color del fondo y Z=Color del texto. Ambos colores son un número del 0 al 15 dado por la tabla que se mostró en el ejemplo anterior.

Así, si el fondo quiero que siga siendo blanco pero que los nombres de los integrantes aparezcan en azul, puedo modificar el código de la siguiente manera:

```
#include<stdlib.h>
...
int main(int argc, char *argv[]) {
    system("color F0");
    logo();
   SetConsoleTextAttribute(GetStdHandle (STD_OUTPUT_HANDLE),241);
   menu_principal();
   return 0;
}
```



❖ *Posicionarse sobre un lugar determinado de la pantalla*

Pueden crear la siguiente función en su programa donde los parámetros “x” e “y” son las coordenadas de la posición donde desean posicionarse.

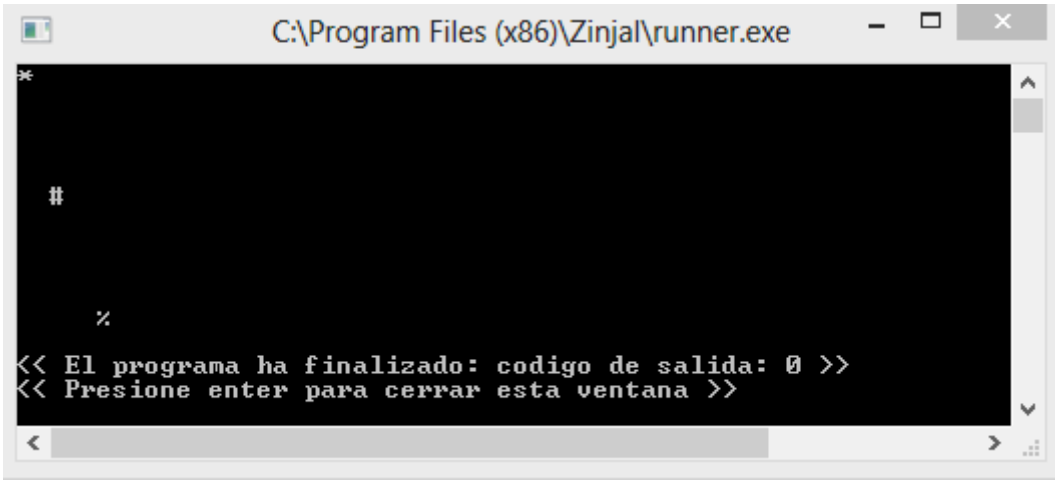
La esquina superior izquierda es el origen de coordenadas.

Tiene que incluirse la libreria <windows.h>

```
#include <windows.h>
...
void gotoxy(USHORT x,USHORT y) {
   COORD cp = {x,y};
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), cp);
}
```

```
1   #include <iostream>
2   #include <cstdlib>
3   #include <windows.h>
4   using namespace std;
5
6   void gotoxy(USHORT x,USHORT y);
7
8   int main(int argc, char *argv[]) {
9       gotoxy(2,5);
10      cout << "#" ;
11      gotoxy(0,0);
12      cout << "*";
13      gotoxy(5,10);
14      cout << "%";
15      return 0;
16  }
17
18  void gotoxy(USHORT x,USHORT y) {
19      COORD cp = {x,y};
20      SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), cp);
21  }
22
```

Ejemplo



- *Realizar una pausa de X cantidad de segundos*

Se puede hacer que el programa detenga una determinada cantidad de milisegundos utilizando la función Sleep() incluida en la librería <windows.h>.

Su uso es bastante sencillo. Lo único que vale la pena aclarar es que el parámetro que se le pasa a la función debe ser un número entero que representa la cantidad de **milisegundos** que dura la pausa. Es decir, si se desea detener todo durante 10 segundos se debería escribir el siguiente código:

```
#include <windows.h>
…
int main(int argc, char *argv[]) {
   ….
   Sleep(10000);
   ….
   return 0;
}
```

❖ *Saber si una tecla fue pulsada anteriormente*

Puede resultar útil crear un algoritmo que realice varias tareas y que, en determinado momento, se pregunte si anteriormente una tecla en particular fue pulsada por el usuario.

La función que sirve para esto es la GetAsynKeyState() que se encuentra en la <windows.h>. Recibe como parámetro la tecla que se quiere evaluar. En el siguiente link pueden encontrar los parámetros para cada tecla:

https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731%28v=vs.85%29.aspx

Al ser invocada, la función devuelve verdadero si anteriormente esa tecla fue pulsada y retorna falso en caso contrario.

Por ejemplo, a continuación se muestra un programa que espera 5 segundos y luego muestra un mensaje dependiendo de si se pulsó la tecla Enter o no:

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

using namespace std;

int main(int argc, char *argv[]) {
    Sleep(5000);
    if(GetAsyncKeyState(VK_RETURN))
        cout << "Se pulso Enter " << endl;
    else
        cout << "No se pulso Enter" << endl;
    system("pause");
    return 0;
}
```

Otra aplicación de gran utilidad sería la siguiente: realizar un bucle infinito que muestre caracteres en pantalla y que solo se detenga al ser pulsada la tecla Escape.

```cpp
#include <iostream>
#include <windows.h>
#include <cstdlib>

using namespace std;

int main(int argc, char *argv[]) {
    while(true){
        cout << "ALGORITMOS" ;
        if(GetAsyncKeyState(VK_ESCAPE)){
            cout << endl << "Fin de bucle" << endl;
            break;
        }
    }
    system("pause");
    return 0;
}
```

```
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
ALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOSALGORITMOS
Fin de bucle
Press any key to continue . . .
```