**Estructura del programa**

*Módulos que contiene:*

int main(void);

void rush01(void);

char \*ft\_strcpy(char \*dest, char \*src);

int input\_control(char input\_text);

void convert\_to\_data(char \*input, int \*input\_data);

void fill\_matrix(int \*input\_data, int \*matrix);

void print\_result(int \*matrix);

void ft\_putchar(char c);

**int main(void)**

{

Llama a la función “rush01”;

Devuelve 0 si todo va bien;

}

**void rush01(void)**

{

Declara un string con nombre “input\_text” de tamaño 33 (16 números, 16 espacios y carácter nulo);

Declara un entero con nombre “input\_ok”;

Declara un array de enteros de dos dimensiones con nombre “input\_data” y tamaño [4][4]. Cuatro filas y cuatro columnas;

Declara un array de enteros de dos dimensiones con el nombre “matrix” y tamaño [4][4]. Cuatro filas y cuatro columnas;

Llama a la función “ft\_strcpy” para rellenar “input\_text”;

Llama a la función “input\_control” para comprobar que los datos de entrada cumplen los requisitos. Si se cumplen (input\_ok == 1):

Llama a la función “convert\_to\_data”, que recorta el string “input\_text”, convierte el texto en números y rellena un array bidimensiona llamado “input\_data” con los datos de entrada;

Llama a la función “fill\_matrix” para rellenar la matriz según la lógica;

Llama a la función “print\_result” para imprimir el resultado en la consola;

Si no se cumplen (input\_ok != 1):

Muestra mensaje de error y finaliza el programa;

}

**char \*ft\_strcpy(char \*dest, char \*src)**

{

Asigna a la variable “input\_text” el string introducido por el segundo parámetro.

(Esta función todavía no sabemos muy bien cómo funciona, ya que ninguno de los tres integrantes del grupo hemos llegado a esos ejercicios aún. Tocaba investigarlo). Sería algo así: input\_text = ft\_strcpy(input\_text, “1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4”);

Devuelve un string;

}

**int input\_control(char \*input\_text)**

{

Recibe por referencia el “input\_text”, comprueba si cumple las restricciones;

Devuelve 1 en caso de cumplirlas o 0 en caso de no cumplirlas;

}

**void convert\_to\_data(char \*input, int \*input\_data)**

{

Recibe por referencia el array “input\_data” y el string “input\_text” Trocea “input\_text” cada 4 posiciones buscando los espacios en blanco, convierte los caracteres en

enteros y los va introduciendo por cada bloque de 4 en el array “input\_data”;

input\_data[0][0] hasta Input\_data[0][3] corresponde a “top columns”.

input\_data[1][0] hasta Input\_data[1][3] corresponde a “bottom columns”.

input\_data[2][0] hasta Input\_data[2][3] corresponde a “left raws”.

input\_data[3][0] hasta Input\_data[3][3] corresponde a “right raws”.

}

**fill\_matrix(int \*input\_data, \*matrix)**

{

Recibe los arrays “input\_data” y “matrix” por referencia. Va comprobando las condiciones de la lógica matemática según los datos de entrada almacenados en “input\_data” y recorriendo la matriz “matrix” todas las veces necesarias para ir rellenándola con el valor que corresponda según las condiciones;

}

**void print\_result(int \*matrix)**

{

Recibe por referencia el array “matrix” y lo va recorriendo;

Por cada posición del array convierte el entero a carácter y llama a la función “ft\_putchar”, que recibe el carácter que corresponde a cada posición y lo imprime por la consola;

}

**void ft\_putchar(char c)**

{

Recibe un carácter por parámetro de entrada y lo escribe en la consola;

}

Resumen de lo que hace el programa:

1. Se reciben unos datos de entrada por parámetro en forma de string y se almacenan en “input\_text”.
2. Se comprueba que no haya símbolos, ni letras, así como otro tipo de restricciones.
3. Los datos están separados por espacios en blanco, por lo que hay que buscar cada dato individual por medio de esos espacios.
4. Recogemos cada bloque de 4, convertimos los caracteres a enteros y los almacenamos en un array de dos dimensiones llamado “input\_data”.
5. Rellena el array “matrix” según las condiciones de la lógica matemática. Habrá que recorrerlo varias veces hasta tenerlo completado. Completado = ninguna posición vacía. O también se puede rellenar al inicio con ceros y entonces completado = ninguna posición con valor 0.
6. Una vez rellenada la matriz completa, imprime el resultado en la consola y finaliza el programa.