//DEFINICIONES

\*\*\* **Iostream**: corriente de entrada y salida

\*\*\* **Arrays:** listas no variables

\*\*\* **Fichero:** contiene la memoria de lo que voy a escribir

\*\*\* **Txt:** fichero de carácter texto

\*\*\* **Endl:** salto de la línea

\*\*\* **fichero.close();**  : para que se pueda guardar todo después y poder cerrar el fichero

\*\*\* **getline**(fichero, mensaje): para escribir una línea, leo el fichero, abre una línea y la mete en el mensaje.

**\*\*\* Eof: end of file,** va al punto del fichero que le toca leer y si lo ultimo que lee es el final de fichero, devuelve 1, porque es el final

\*\*\* **Char:** es una letra, variable tipo letra

\*\*\* **String:** conjunto de letras.

\*\*\* **Struct:**  indica se está definiendo una estructura.

\*\*\* **ifstream fichero("");**  'el comando ifstream busca la ruta que escriba, (lo lee del disco duro y lo vuelca, lo escribe en la memoria ram)

\*\*\* **ofstream:** corresponde al fichero de salida

\*\***\* getline(fichero**); ' va la memoria ram y lee todos los símbolos que hay hasta que encuentre el símbolo de retorno de carro.  Guarda la línea de texto en una variable, que llamaremos línea.

\*\*\* **PUNTEROS:** cuando hay que reservar espacio de la RAM se hace una variable. Se **llaman punteros porque apuntan a otro lugar. PX es para apuntar a X.**

**\*\*\* EL MÉTODO DE ORDENACION POR BURBUJA: (método clásico)**

(se pueden ordenar números, nombres…)

Se cogen dos elementos y se plantea si están bien ordenados, la respuesta puede ser sí o no, si es no, se invierten los números.

Se cogen otros dos elementos y se vuelven a invertir si no están ordenados. Con esto he conseguido que algunos de los números se desplacen pero no está ordenado entero, para ello se vuelve a comenzar el proceso. Hasta que se ordenen los elementos.

**¿Por qué se llama burbuja?** : Cuando se localiza un número muy grande mal colocado ese número sube a la superficie como las burbujas del champan.

\*\*\* **Int 'a' :**Esto es un variable llamada 'a' que va a guardar una variable de tipo entero

**%i** es un número entero

**%c** es la letra

**%x** es un número hexadecimal

**%s** es una cadena de caracteres

**%f** es un numero decimal (float)

\*\*\* **Malloc:** reserva de espacio, para buscar el tamaño de memoria que necesita

\*\*\* **strcpy:** copia el contenido de la <cadena\_fuente> en la <variable\_destino>, siendo ésta un array de caracteres. Manera rápida de copiar lo que pone en la derecha en la dirección izquierda.

\*\*\***\*;** Contenido

\*\*\*&; Dirección