

Examen Práctico de Programación 1 - 22/enero/2016

- Tiempo máximo para realizar el trabajo de programación propuesto: 90 minutos
- Entrega del trabajo a través de la plataforma *Moodle2*.

Especificación del trabajo a desarrollar en el turno 1º (15:00 horas)

Se ha de desarrollar un programa, en el área de trabajo **trabajos**, ubicada en la carpeta **practicaprogramacion1**, que presente el siguiente comportamiento:

```
Fichero de subtítulos (de practicasPROGRAMACION1/datos/subtitulos): sub01.srt
```

```
El fichero contiene 1510 subtítulos.
```

```
Los 5 subtítulos mas breves del fichero:
```

```
NUM.      DURACION
====      =====
0576 00:00:00,710
1258 00:00:00,760
0234 00:00:00,770
0507 00:00:00,770
0160 00:00:00,800
```

```
Los 5 subtítulos mas largos del fichero:
```

```
NUM.      DURACION
====      =====
1162 00:00:05,760
0391 00:00:05,840
0083 00:00:05,880
1166 00:00:07,360
1212 00:00:08,560
```

```
Todos los subtítulos han sido almacenados ordenados en dur_sub01.srt
```

El programa ha de realizar las tareas que se describen a continuación:

- En primer lugar pide al operador que escriba el nombre de un fichero de subtítulos (en formato **SubRip**) que ha de estar almacenado en la carpeta **practicaprogramacion1/datos/subtitulos** y lee el nombre del fichero escrito por el operador (**sub01.srt**, en el ejemplo anterior).
- Informa del número de subtítulos que almacena el fichero (1510 subtítulos en el ejemplo anterior).
- Ordena los subtítulos de menor a mayor duración, en tiempo, e informa de los cinco subtítulos del fichero cuya duración sea menor (los subtítulos número 576, 1258, 234, 507 y 160 en el ejemplo anterior).
- Informa de los cinco subtítulos del fichero cuya duración, en tiempo, sea mayor (los subtítulos número 1162, 391, 83, 1166 y 1212 en el ejemplo anterior).
- Informa que todos los subtítulos, tras haber sido ordenados de menor a mayor duración, en tiempo, han sido almacenados en un nuevo fichero, **dur_sub01.srt**, que los almacena en formato **SubRip**. Este fichero se almacena también en la carpeta **practicaprogramacion1/datos/subtitulos**. Su nombre es

el resultado de anteponer el prefijo *dur_* al nombre del fichero inicial *sub01.srt*. Su contenido se muestra parcialmente a continuación.

```
576
00:37:28,730 --> 00:37:29,440
This morning's front page?

1258
01:28:57,000 --> 01:28:57,760
Come in.

. . .

1166
01:20:08,000 --> 01:20:15,360
What's it called, Shangri-la? El Dorado?
Sloppy Joe's? What's the name?

1212
01:23:22,000 --> 01:23:30,560
Get the voice out of the throat
Place the tone right in the mask
```

El diseño del programa anterior se simplifica notablemente declarando datos de los tipos **Tiempo** y **Subtitulo**, definidos en los módulos **tiempo** y **subtitulo**, respectivamente, y haciendo uso de las funciones visibles definidas en los módulos **tiempo**, **subtitulo** y **herramientas**.

El programa constará únicamente de un módulo principal, que hará uso intensivo de los recursos definidos en los tres módulos que acaban de ser citados.

Se debe entregar, a través de la plataforma **Moodle2**, un único fichero, el que almacene el código fuente del módulo principal del programa y cuyas primeras líneas sean un comentario inicial con el nombre y apellidos del autor del trabajo.

Esta prueba de programación se valorará según los siguientes criterios:

- El programa informa correctamente del número de subtítulos del fichero y presenta el listado de los cinco subtítulos de menor duración y otro de los cinco subtítulos de mayor duración [3 puntos]
- El programa ha creado un nuevo fichero, con el nombre especificado, que almacena ordenados todos los subtítulos en orden de duraciones crecientes y en formato **SubRip** [3 puntos]
- Diseño y legibilidad del código del módulo principal del programa [4 puntos]
- Un programa que presente errores al ser compilado o presente errores graves al ser ejecutado será valorado con 0 puntos.

Examen Práctico de Programación 1 - 22/enero/2016

- Tiempo máximo para realizar el trabajo de programación propuesto: 90 minutos
- Entrega del trabajo a través de la plataforma *Moodle2*.

Especificación del trabajo a desarrollar en el turno 2º (17:00 horas)

Se ha de desarrollar un programa, en el área de trabajo **trabajos**, ubicada en la carpeta **practicaprogram1**, que presente el siguiente comportamiento:

Fichero de subtítulos (en practicasPROG1/datos/subtitulos): **sub02.srt**

El fichero contiene 765 subtítulos.

Mínimo y máximo número de caracteres por subtítulo: **70 100**

Subtítulos con un número de caracteres entre 70 y 100:

NUM.	CARACTERES
====	=====
0348	70
0139	70
0422	71
0166	72
0472	73
0361	73
0159	74
0646	77

Los subtítulos seleccionados y ordenados han sido almacenados en `numcar_sub02.srt`

El programa ha de realizar las tareas que se describen a continuación:

- En primer lugar pide al operador que escriba el nombre de un fichero de subtítulos (en formato **SubRip**) que ha de estar almacenado en la carpeta **practicaprogram1/datos/subtitulos** y lee el nombre del fichero escrito por el operador (**sub02.srt**, en el ejemplo anterior).
- Informa del número de subtítulos que almacena el fichero (765 subtítulos en el ejemplo anterior).
- Pide al operador que defina el número mínimo y máximo de caracteres que han de tener los subtítulos que se van a listar a continuación (70 y 100 caracteres, respectivamente, en el ejemplo anterior).
- Ordena todos los subtítulos según número de caracteres creciente y presenta por pantalla un listado que informa del número de los subtítulos cuyo número de caracteres está incluido en el intervalo definido por el operador (intervalo [70,100] en el ejemplo anterior) y del número de caracteres de cada uno de ellos. En el listado los subtítulos se presentan ordenados de menor a mayor número de caracteres.
- Informa que los subtítulos cuya longitud, en caracteres, está comprendida entre los dos valores determinados por el operador, han sido almacenados, ordenados según número de caracteres creciente, en un nuevo fichero, **numcar_sub02.srt**, con formato **SubRip**. Este fichero se almacena

también en la carpeta *practicaprogramacion/datos/subtitulos*. Su nombre es el resultado de anteponer el prefijo *numcar_* al nombre del fichero inicial *sub02.srt*. Su contenido se muestra parcialmente a continuación.

```
348
00:24:46,606 --> 00:24:50,182
What I hear, he's already got more men
than any of your southern kings.

139
00:09:25,788 --> 00:09:28,846
We have nowhere to house them.
And with winter coming, it'll get worse.

. . .

159
00:10:33,549 --> 00:10:37,430
I do believe the Hand of the King is
welcome at all small council meetings.

646
00:44:50,496 --> 00:44:53,699
- Shall we rally them to fight for us?
- I understand you don't trust Greyjoy.
```

Las situaciones no especificadas las debe resolver cada alumno aplicando el sentido común y pensando siempre en la facilidad de uso del programa.

El diseño del programa anterior se simplifica notablemente declarando datos de los tipos **Tiempo** y **Subtitulo**, definidos en los módulos **tiempo** y **subtitulo**, respectivamente, y haciendo uso de las funciones visibles definidas en los módulos **tiempo**, **subtitulo** y **herramientas**.

El programa requerirá únicamente la escritura de un módulo principal, que hará uso intensivo de los recursos definidos en los tres módulos de biblioteca que acaban de ser citados.

Se debe entregar, a través de la plataforma **Moodle2**, un único fichero, el que almacene el código fuente del módulo principal del programa y cuyas primeras líneas sean un comentario inicial con el nombre y apellidos del autor del trabajo.

Esta prueba de programación se valorará según los siguientes criterios:

- El programa informa correctamente del número de subtítulos del fichero y presenta un listado ordenado de los subtítulos cuya longitud, en número de caracteres, esté comprendida entre los límites definidos previamente por el operador [3 puntos]
- El programa ha creado un nuevo fichero, con el nombre especificado, en el que almacena los subtítulos listados anteriormente en orden creciente de su número de caracteres y en formato **SubRip** [3 puntos]
- Diseño y legibilidad del código del módulo principal del programa [4 puntos]
- Un programa que presente errores al ser compilado o presente errores graves al ser ejecutado será valorado con 0 puntos.

Una solución del problema propuesto en el turno 1º

```
/*
 * Programadores del módulo: Javier Martínez y Miguel Ángel Latre
 * Fecha de la última revisión : 5/noviembre/2015
 */

#include <iostream>
#include <iomanip>
#include <cstring>
#include <fstream>

using namespace std;

#include "../ModTiempo/tiempo.h" // el módulo tiempo
#include "../ModSubtitulo/subtitulo.h" // el módulo subtitulo
#include "../ModHerramientas/herramientas.h" // el módulo herramientas

/*
 * Pre: TS[0,n-1] almacena n subtítulos desde >= 1 y hasta <= n
 * Post: Presenta por pantalla una tabla con los números y duraciones de
 * los subtítulos TS[desde,hasta]. Un ejemplo:
 * NUM. DURACION
 * =====
 * 0167 01:01:14,660
 * 1109 00:23:04,900
 * . . .
 * 0446 00:45:00,000
 */
void presentar (const Subtitulo TS[], const int desde, const int hasta) {
    if (desde <= hasta) {
        // Escribe la cabecera de la tabla
        cout << "NUM. .... DURACION" << endl;
        cout << "===== " << endl;
        for (int i = desde; i <= hasta; ++i) {
            // Escribe una línea con el número y la duración del subtítulo i-ésimo,
            // es decir, de TS[i-1]
            cout << setfill('0') << setw(4) << numero(TS[i-1]) << " ";
            mostrar( restar ( fin (TS[i-1]), principio (TS[i-1])));
            cout << endl;
        }
    }
}

/*
 * Pre: n >= 0
 * Post: Almacena los subtítulos TS[0,n-1] en el fichero nombreFichero
 * con formato 'subRip' y en el mismo orden: TS[0], TS[1], etc.
 */
void guardar (const Subtitulo TS[], const int n,
              const char nombreFichero[]) {
    ofstream f; // Declara un flujo de datos de salida
    // Asocia al flujo f el fichero nombreFichero para su creación
    f.open (nombreFichero);
    if (f.is_open ()) {
        for (int i = 0; i < n; ++i) {
            // Escribe el número del subtítulo TS[i]
            f << numero(TS[i]) << endl;
            // Escribe los tiempos de comienzo y de final del subtítulo TS[i]
            Tiempo t = principio (TS[i]);
            f << setfill('0') << setw(2) << hora(t) << ':' << setw(2) << minuto(t) << ':'

```

```

        << setw(2) << segundo(t) << ',' << setw(3) << milesima(t);
    f << " _-->_";
    t = fin (TS[i ]);
    f << setw(2) << hora(t) << ':' << setw(2) << minuto(t) << ':'
        << setw(2) << segundo(t) << ',' << setw(3) << milesima(t) << endl;
    // Escribe el texto del subtítulo TS[i]
    int nLineas;
    char contenido[MAX_LINEAS][MAX_LONG_LINEA];
    texto (TS[i ], nLineas, contenido );
    for (int j = 1; j <= nLineas; ++j) {
        f << contenido[j-1] << endl;
    }
    // Escribe una línea sin caracteres (fin del subtítulo )
    f << endl;
}
}
}
}

/*
 * Este programa realiza las siguientes tareas:
 * 1. Pide el operador que defina el nombre de un fichero de subtítulos con formato
 * SubRip, que ha de estar ubicado en la carpeta RUTA, y lee su respuesta. Ej:
 * Un fichero de subtítulos (de practicasPROG1/datos/ subtítulos ): s01. srt
 * 2. Informa del número de subtítulos almacenados en el fichero
 * 3. Informa de los números y duraciones de sus cinco subtítulos con menor duración
 * 4. Informa de los números y duraciones de sus cinco subtítulos con mayor duración
 * 5. Crea en la carpeta RUTA un nuevo fichero cuyo nombre sea el mismo que el inicial ,
 * con el prefijo "dur_", y almacena en él todos los subtítulos ordenados según
 * duraciones crecientes e informa de ello:
 * Los subtítulos ordenados han sido almacenados en dur_s01. srt
 */
int main () {
    // Carpeta que almacena los ficheros de subtítulos
    const char RUTA[] = " ../././ datos/ subtítulos /";
    // Nombre de fichero (sin ruta) y con ruta
    char nombreFichero[64];
    char nombreCompletoFichero[128];
    // Pregunta por el nombre de un fichero de subtítulos
    cout << "Un fichero de subtítulos (en practicasPROG1/datos/ subtítulos ): " << flush;
    cin >> nombreFichero;
    strcpy (nombreCompletoFichero, RUTA);
    strcat (nombreCompletoFichero, nombreFichero);
    // Asigna a numSubtitulos el número de subtítulos del fichero nombreCompletoFichero
    int numSubtitulos = contarSubtitulos (nombreCompletoFichero);
    if (numSubtitulos > 0) {
        // Informa del número de subtítulos del fichero
        cout << endl << "El fichero contiene " << numSubtitulos << " subtítulos " << endl
            << endl;
        // Lee los subtítulos del fichero nombreCompletoFichero y los almacena en
        // TS[0,numSubtitulos-1]
        const int MAXIMO = 2000; // Núm. máximo de subtítulos que pueden tratarse
        Subtitulo TS[MAXIMO]; // Tabla de subtítulos
        leerSubtitulos (nombreCompletoFichero, TS);
        // Ordena los subtítulos de TS[0,numSubtitulos-1] según duración creciente
        ordenar(TS, numSubtitulos, 'D');
        // Presenta por pantalla los 5 subtítulos de menor duración
        cout << "Los 5 subtítulos mas breves del fichero : " << endl << endl;
        presentar (TS, 1, 5);
        // Presenta por pantalla los 5 subtítulos de mayor duración
        cout << endl << "Los 5 subtítulos mas largos del fichero : " << endl << endl;
        presentar (TS, numSubtitulos-4, numSubtitulos);
    }
}

```

```

cout << endl;
// Almacena los subtítulos TS[0,numSubtitulos-1] en un nuevo fichero
const char PREFIJO[] = "dur.";
strcpy (nombreCompletoFichero, RUTA);
strcat (nombreCompletoFichero, PREFIJO);
strcat (nombreCompletoFichero, nombreFichero);
guardar(TS, numSubtitulos, nombreCompletoFichero);
// Informa del nombre del fichero que almacena los subtítulos ordenados
cout << "Los subtítulos ordenados han sido almacenados en"
<< PREFIJO << nombreFichero << endl;
}
else {
    cout << "No se ha podido leer el fichero" << nombreFichero << endl;
}
return 0;
}

```

Una solución del problema propuesto en el turno 2º

```
/*
 * Programadores del módulo: Javier Martínez y Miguel Ángel Latre
 * Fecha de la última revisión : 5/noviembre/2015
 */

#include <iostream>
#include <iomanip>
#include <cstring>
#include <fstream>

using namespace std;

#include "../ModTiempo/tiempo.h" // el módulo tiempo
#include "../ModSubtitulo/subtitulo.h" // el módulo subtitulo
#include "../ModHerramientas/herramientas.h" // el módulo herramientas

/*
 * Pre: ----
 * Post: Devuelve el número total de caracteres del subtitulo S
 */
int caracteresSubtitulo (const Subtitulo S) {
    char unTexto[MAX_LINEAS][MAX_LONG_LINEA];
    int lineasSubtitulo ;
    int longitud = 0;
    texto(S, lineasSubtitulo , unTexto);
    for (int linea = 0; linea < lineasSubtitulo ; ++linea) {
        longitud = longitud + strlen (unTexto[ linea ]);
    }
    return longitud ;
}

/*
 * Pre: TS[0,n-1] almacena n subtítulos desde >= 1 y hasta <= n
 * Post: Presenta por pantalla un listado con el número de subtítulo y el
 * número de caracteres de los subtítulos TS[0,n] cuyo número de
 * caracteres esté comprendido en el intervalo [minimo,maximo]. El orden
 * de presentación respetará el orden de almacenamiento en la tabla (se
 * comienza por los subtítulos de índices más bajos y se acaba por los
 * de índices más altos . Un ejemplo:
 * NUM. CARACTERES
 * =====
 * 1209 70
 * 0237 72
 * . . .
 * 0444 92
 * 1012 92
 */
void presentar (const Subtitulo TS[], const int n, const int minimo, const int maximo) {
    if (minimo <= maximo) {
        // Escribe la cabecera de la tabla
        cout << "NUM. ....CARACTERES" << endl;
        cout << "=====_" << endl;
        bool fin = false ;
        for (int i = 0; !fin && i < n; ++i) {
            int numChar = caracteresSubtitulo (TS[i]);
            if (numChar >= minimo) {
                if (numChar <= maximo) {
                    // Escribe una línea con el número y el número de caracteres
                    // del subtítulo TS[i]
                }
            }
        }
    }
}
```



```

        cout << setfill ('0') << setw(4) << numero(TS[i]) << "___"
            << setfill (' ') << setw(7) << numChar << endl;
    }
    else {
        fin = true;
    }
}
}
}
}
}
}
}
}
}
}

/*
 * Pre: n >= 0
 * Post: Almacena los subtítulos TS[0,n-1] cuya longitud este comprendida
 * entre min caracteres y maxcaracteres en el fichero nombreFichero
 * con formato 'subRip' y en el mismo orden en que están dispuestos
 * en TS[0,n-1]
 */
void guardar (const Subtitulo TS[], const int n, const char nombreFichero[],
              const int min, const int max) {
    ofstream f; // Declara un flujo de datos de salida
    // Asocia al flujo f el fichero nombreFichero para su creación
    f.open (nombreFichero);
    if (f.is_open ()) {
        for (int i = 0; i < n; ++i) {
            // Calcula el número de caracteres del subtítulo TS[i]
            int numCar = caracteresSubtitulo (TS[i]);
            if (numCar >= min && numCar <= max) {
                // Escribe el número del subtítulo TS[i]
                f << numero(TS[i]) << endl;
                // Escribe los tiempos de comienzo y de final del subtítulo TS[i]
                Tiempo t = principio (TS[i]);
                f << setfill ('0') << setw(2) << hora(t) << ':' << setw(2) << minuto(t) << ':'
                    << setw(2) << segundo(t) << ',' << setw(3) << milesima(t);
                f << "___>_";
                t = fin (TS[i]);
                f << setw(2) << hora(t) << ':' << setw(2) << minuto(t) << ':'
                    << setw(2) << segundo(t) << ',' << setw(3) << milesima(t) << endl;
                // Escribe el texto del subtítulo TS[i]
                int nLineas;
                char contenido[MAX_LINEAS][MAX_LONG_LINEA];
                texto (TS[i], nLineas, contenido);
                for (int j = 1; j <= nLineas; ++j) {
                    f << contenido[j-1] << endl;
                }
                // Escribe una línea sin caracteres (fin del subtítulo)
                f << endl;
            }
        }
    }
}

/*
 * Este programa realiza las siguientes tareas:
 * 1. Pide el operador que defina el nombre de un fichero de subtítulos con formato
 * SubRip, que ha de estar ubicado en la carpeta RUTA, y lee su respuesta. Ej:
 * Un fichero de subtítulos (de practicasPROG1/datos/subtitulos): s02.srt
 * 2. Informa del número de subtítulos almacenados en el fichero
 * 3. Pide al operador que defina un número mínimo y un número máximo de
 * caracteres por subtítulo
 * 4. Presenta un listado de subtítulos, limitado a su número de subtítulo y

```

```

*   su longitud (en número de caracteres ), cuyo número de caracteres esté
*   comprendido entre el mínimo y el máximo definido previamente por el
*   operador. El listado se presenta por pantalla ordenado según número
*   de caracteres creciente
* 5. Crea en la carpeta RUTA un nuevo fichero cuyo nombre sea el mismo que el inicial ,
*   con el prefijo "numcar_", y almacena en él los subtítulos listados previamente
*   y ordenados según número de caracteres creciente e informa de ello :
*   Los subtítulos ordenados han sido almacenados en numcar_s02.srt
*/
int main () {
    // Carpeta que almacena los ficheros de subtítulos
    const char RUTA[] = " ../ ../ datos/ subtítulos /";
    // Nombre de fichero (sin ruta) y con ruta
    char nombreFichero[64];
    char nombreCompletoFichero[128];
    // Pregunta por el nombre de un fichero de subtítulos
    cout << "Un_fichero_de_subtitulos_(en_practicaprogram1/datos/ subtítulos ): " << flush;
    cin >> nombreFichero;
    strcpy (nombreCompletoFichero, RUTA);
    strcat (nombreCompletoFichero, nombreFichero);
    // Asigna a numSubtitulos el número de subtítulos del fichero nombreCompletoFichero
    int numSubtitulos = contarSubtitulos (nombreCompletoFichero);
    if (numSubtitulos > 0) {
        // Informa del número de subtítulos del fichero
        cout << endl << "El_fichero_contiene_" << numSubtitulos << "_subtitulos" << endl
            << endl;
        // Pregunta por el mínimo y el máximo número de caracteres de un subtítulo
        // a considerar
        int minimo, maximo;
        cout << "Mínimo_y_máximo_número_de_caracteres_por_subtítulo:_" << flush;
        cin >> minimo >> maximo;
        // Lee los subtítulos del fichero nombreCompletoFichero y los almacena en
        // TS[0,numSubtitulos-1]
        const int MAXIMO = 2000; // Núm. máximo de subtítulos que pueden tratarse
        Subtitulo TS[MAXIMO]; // Tabla de subtítulos
        leerSubtitulos (nombreCompletoFichero, TS);
        // Ordena los subtítulos de TS[0,numSubtitulos-1] según número de
        // caracteres creciente
        ordenar(TS, numSubtitulos, 'L');
        // Presenta por pantalla los subtítulos cuyo número de caracteres esté
        // comprendido entre minimo y maximo, ordenados según número de caracteres
        // creciente
        cout << endl << "Subtitulos_con_un_número_de_caracteres_entre_"
            << minimo << "_y_" << maximo << ":" << endl << endl;
        presentar (TS, numSubtitulos, minimo, maximo);
        cout << endl;
        // Almacena los subtítulos de TS[0,numSubtitulos-1], cuyo número de
        // caracteres esté comprendido entre minimo y maximo, en un nuevo
        // fichero , ordenados según número de caracteres creciente
        const char PREFIJO[] = "numcar.";
        strcpy (nombreCompletoFichero, RUTA);
        strcat (nombreCompletoFichero, PREFIJO);
        strcat (nombreCompletoFichero, nombreFichero);
        guardar(TS, numSubtitulos, nombreCompletoFichero, minimo, maximo);
        // Informa del nombre del fichero que almacena los subtítulos ordenados
        cout << "Los_subtitulos_ordenados_han_sido_almacenados_en_"
            << PREFIJO << nombreFichero << endl;
    }
    else {
        cout << "No_se_ha_podido_leer_el_fichero_" << nombreFichero << endl;
    }
}

```

```
return 0;  
}
```