

O.1 – Planificación de horarios (B)

Estructuras de Datos
Facultad de Informática - UCM

Supongamos que queremos planificar las actividades de un congreso. Cada actividad tiene una duración diferente. Para almacenar la duración de cada una de ellas utilizamos el siguiente tipo de datos:

```
using Duracion = int;
```

Si no conoces la palabra clave `using` en C++, posiblemente sí conozcas `typedef`. La declaración anterior es equivalente a lo siguiente:

```
typedef int Duracion;
```

En cualquier caso, expresamos cada duración mediante un número entero que expresa una cantidad de segundos determinada.

Por otro lado, el congreso también tiene una duración máxima, y queremos saber si va a haber tiempo suficiente para poder abarcar todas las actividades propuestas. Vamos a suponer que tenemos un listado con las duraciones de cada una de estas actividades, una por línea, y cada una con el formato HH:MM:SS, donde HH indica el número de horas, MM indica el número de minutos y SS indica el número de segundos. No indicamos el número de días porque suponemos que todas las actividades duran menos de un día. El listado finaliza con la duración 00:00:00.

Implementa la siguiente función:

```
bool caben_todas(const Duracion &duracion_congreso);
```

Esta función recibe la duración máxima del congreso, y debe leer de la entrada el listado de duraciones de las actividades. La función debe devolver `true` si el tiempo ocupado por todas las actividades es estrictamente menor que la duración del congreso, o `false` en caso contrario.

Solución

```
#include <iostream>

using namespace std;

using Duracion = int;

const int LONG_STRING = 11;

Duracion from_string(const string &duracion) {
    int dias, horas, minutos, segundos;
    if (duracion.length() == LONG_STRING) {
        dias = stoi(duracion.substr(0, 2));
        horas = stoi(duracion.substr(3, 2));
        minutos = stoi(duracion.substr(6, 2));
        segundos = stoi(duracion.substr(9, 2));
    } else {
        dias = 0;
        horas = stoi(duracion.substr(0, 2));
        minutos = stoi(duracion.substr(3, 2));
        segundos = stoi(duracion.substr(6, 2));
    }

    return dias * 86400 + horas * 3600 + minutos * 60 + segundos;
}

Duracion sumar_duracion(const Duracion &t1, const Duracion &t2) {
    return t1 + t2;
}

bool duracion_menor_que(const Duracion &t1, const Duracion &t2) {
    return t1 < t2;
}

Duracion duracion_cero() {
    return 0;
}

bool caben_todas(const Duracion &duracion_congreso) {
    string hora_str;
    cin >> hora_str;
```

```
Duracion actual = duracion_cero();

while (hora_str != "00:00:00") {
    actual = sumar_duracion(actual, from_string(hora_str));
    cin >> hora_str;
}

return duracion_menor_que(actual, duracion_congreso);
}
```