

Programación 2

Examen de teoría (julio 2016)

8 de julio de 2016



Instrucciones

- **Duración: 3 horas**
- El fichero del primer problema debe llamarse `quijote.cc`. Para el segundo problema es necesario entregar cuatro ficheros, llamados `Partido.cc`, `Partido.h`, `Provincia.cc`, `Provincia.h`. Pon tu DNI y tu nombre en un comentario al principio de los ficheros fuente.
- La entrega se realizará como en las prácticas, a través del servidor del DLSI (<http://pracdlsi.dlsi.ua.es>), en el enlace Programación 2. Puedes realizar varias entregas, aunque sólo se corregirá la última.
- En la web de la asignatura <http://www.dlsi.ua.es/asignaturas/p2> tienes disponibles algunos ficheros que te pueden servir de ayuda para resolver el examen, y el apartado Reference de la web www.cplusplus.com.

Problemas

1. (4.5 puntos)

Este año se cumple el cuarto centenario de la muerte de Miguel de Cervantes. Como homenaje al autor, te has propuesto leer el Quijote y recopilar en un fichero las mejores citas de esta obra culmen de las letras hispánicas. Además, para cada una de las citas quieres almacenar el capítulo en el que aparece¹ y una puntuación, entre 1 y 10 (ambos incluidos), indicando tu valoración de la misma.

Para poder visualizar toda esta información de manera adecuada una vez que termines de leer el libro, has decidido realizar un programa que reciba como parámetro de entrada ese fichero de citas, las agrupe por capítulos y las muestre por pantalla. A continuación se muestra un ejemplo de fichero de entrada:²

```
En un lugar de la Mancha, de cuyo nombre no quiero acordarme$1$5
Enderecese ese bacín que trae en la cabeza, y no ande buscando tres pies al gato$21$3
Sin duda alguna, eran molinos de viento, y no gigantes, aquellos que iba a acometer$6$7
Oh hildeputa, qué reja que tiene, y qué voz!$6$9
Don Ginesillo de Paropillo, o como os llamáis, que habéis de ir vos solo, rabo entre piernas$21$8
Comenzó el pobre escudero a desaguarse por entrambas canales$17$1
No ha mucho tiempo que vivía un hidalgo de los de lanza en astillero$1$2
```

En cada línea del fichero se incluye, separado por el símbolo \$, el texto de la cita, el número de capítulo en el que aparece y la puntuación asignada por ti.

El programa recibirá como parámetro de entrada el nombre del fichero. Opcionalmente se le podrá pasar también un número entero que indique un umbral de corte para la puntuación de las citas que se van a mostrar. Es decir, para una llamada al programa como `quijote citas.txt 3` el programa sólo mostrará aquellas citas contenidas en el fichero `citas.txt` cuya puntuación sea igual o mayor a 3. Si no se indica ningún valor en los parámetros, se mostrarán todas las citas independientemente de su puntuación. Si el número indicado no está entre 1 y 10 se debe mostrar un error y terminar el programa.

Un ejemplo de salida para la llamada `quijote citas.txt 3` y el ejemplo anterior sería:

```
Capítulo 1
- En un lugar de la Mancha, de cuyo nombre no quiero acordarme
Capítulo 21
- Enderecese ese bacín que trae en la cabeza, y no ande buscando tres pies al gato
- Don Ginesillo de Paropillo, o como os llamáis, que habéis de ir vos solo, rabo entre piernas
Capítulo 6
- Sin duda alguna, eran molinos de viento, y no gigantes, aquellos que iba a acometer
- Oh hildeputa, qué reja que tiene, y qué voz!
```

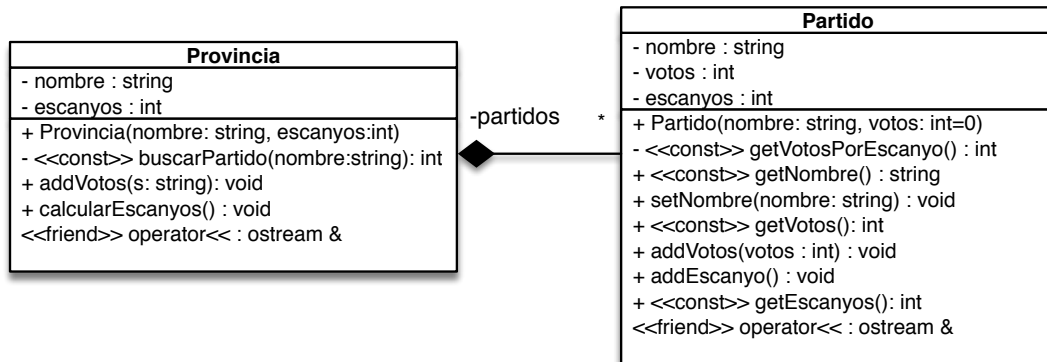
¹En realidad habría que indicar también en cuál de las dos partes del libro aparece, pero aquí obviaremos ese detalle.

²Asumiremos que el contenido del fichero es siempre correcto y que los datos son válidos. No hay que comprobarlo.

No hace falta ordenar los capítulos a la hora de mostrarlos. Si para un capítulo no hay ninguna cita que supere el umbral, no se debe mostrar nada para ese capítulo (como sucede con el capítulo 17 en el ejemplo anterior).

2. (5.5 puntos)

Se pide implementar un programa que gestione los resultados de las elecciones por provincias, asignando un número de escaños a cada partido según la ley D'Hont.



Un **Partido** tiene un nombre (por ejemplo, “PSOE”), un número de votos y de escaños. El constructor debe inicializar los datos y lanzar una excepción si el número de votos es negativo. El método `getVotosPorEscanyo` devuelve el resultado de la división entre ambos, o 0 si no hay ningún escaño. El método `addVotos` añade el número de votos pasados por parámetro. Si este número es menor que cero debe lanzar una excepción. El método `addEscanyo` añade un nuevo escaño al partido. El operador salida debe escribir los datos de un partido como se muestra en el ejemplo del final.

Una **Provincia** tiene un nombre (por ejemplo, “Alicante”), y el número total de escaños que le corresponden. El método `buscarPartido` debe buscar un partido por su nombre en el vector de partidos, y devolver su posición o `-1` si no está. El método `addVotos` debe actualizar los votos del partido correspondiente, añadiéndolo al vector si no estaba. Para ello recibe un `string` con el nombre del partido, un espacio y el número de votos³. Es posible que se llame al método `addVotos` varias veces con el mismo partido, si por ejemplo se cuentan mesas electorales distintas.

El método `calcularEscanyos` calcula la ley D'Hont, asignando los escaños en función de los votos. Para esto debe construirse una matriz con tantas filas como partidos, y tantas columnas como escaños a repartir (N). A continuación tenemos que rellenar esta matriz. En cada fila ponemos un partido, y las columnas deben guardar el número total de votos de dicho partido dividido por su índice en la tabla. Por ejemplo, la primera columna tendría los votos totales del partido correspondiente, la segunda estos mismos votos divididos⁴ entre 2, la tercera los votos divididos entre 3, y así hasta N . Una vez construida la tabla se cogen los N mayores valores de la misma, y se asigna cada uno de estos escaños a los partidos de las filas a las que corresponden.

Dado el siguiente `main.cc` (que puedes compilar con `g++ Partido.cc Provincia.cc main.cc -o ej2`):

```

#include "Provincia.h"
int main()
{
    Provincia p("Alicante",12);

    p.addVotos("PP 300000");
    p.addVotos("UP-Compromis 192165");
    p.addVotos("PSOE 187056");
    p.addVotos("PP 9000"); // Otro colegio electoral
    p.addVotos("Cs 138063");
    p.addVotos("PP -100"); // Excepción lanzada
    p.addVotos("PACMA 10600");
    p.addVotos("PP 20000");

    p.calcularEscanyos();
    cout << p << endl;
}
  
```

el programa debe imprimir:

```

Valor -100 incorrecto
PP: 5 escanyos, 329000 votos, 65800 votos por escanyo
UP-Compromis: 3 escanyos, 192165 votos, 64055 votos por escanyo
PSOE: 2 escanyos, 187056 votos, 93528 votos por escanyo
Cs: 2 escanyos, 138063 votos, 69031 votos por escanyo
PACMA: 0 escanyos, 10600 votos
  
```

³Se supone que la sintaxis de este `string` será siempre correcta.

⁴La división debe ser entera.