

Programación 2

Examen de teoría (julio 2011)

8 de julio de 2011



Instrucciones

- **Duración: 3 horas**
- El fichero del primer ejercicio debe llamarse `ej1.cc`. Para el segundo problema es necesario entregar cuatro ficheros, llamados `ajedrez.cc`, `ajedrez.h`, `pieza.cc`, `pieza.h`. Pon tu DNI y tu nombre en un comentario al principio de los ficheros fuente.
- La entrega se realizará a través del servidor de prácticas del DLSI (<http://pracdlsi.dlsi.ua.es>), del mismo modo que si fuera una práctica. Puedes realizar varias entregas, aunque sólo se corregirá la última.

Problemas

1. (5.5 puntos)

Implementa un programa que reciba como argumentos¹ el nombre de un fichero de texto e imprima por pantalla sus líneas junto con el número de palabras distintas en cada una de ellas. Se considerará que una palabra es una secuencia de una o más letras consecutivas. Por ejemplo, si el fichero de entrada contiene lo siguiente:

```
En un lugar de la Mancha de cuyo nombre...
Uno, DOS, dos, Hola!, hola.
```

El programa debería imprimir:

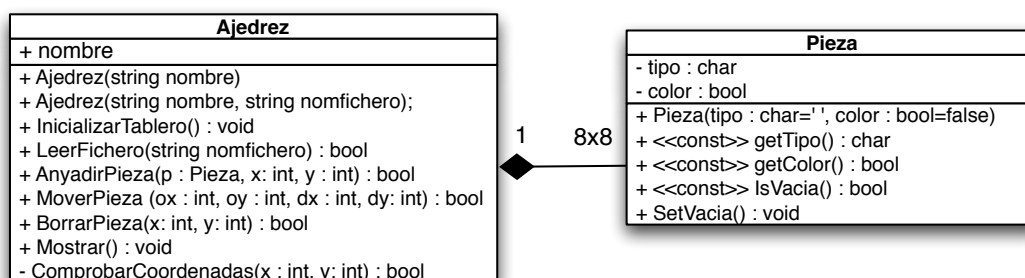
```
8 En un lugar de la Mancha de cuyo nombre...
3 Uno, DOS, dos, Hola!, hola.
```

Para comparar palabras, no se deberá tener en cuenta si sus letras son mayúsculas o minúsculas². Se supondrá que no habrá más de 100 palabras distintas en cada línea.

Nota: Este problema debe implementarse usando programación procedural (no orientada a objetos).

2. (4.5 puntos)

Queremos hacer un programa para visualizar un tablero de ajedrez y mover sus piezas. Para ello, partimos del siguiente diagrama de clases³:



¹Hay que comprobar que el usuario introduce un argumento y en caso contrario mostrar un mensaje de error.

²Puedes convertir una letra mayúscula en minúscula así: `char letramin = letramay-'A'+'a';`

³La relación de composición se implementará mediante una matriz de 8x8 piezas.

Las piezas pueden ser de tipo peón (p), torre (t), caballo (c), alfil (a), rey (r) o dama (d), y pueden ser de color blanco (b) o negro (n). Por ejemplo, teniendo el siguiente ejemplo de fichero `ajedrez.txt`, que contiene una línea por pieza indicando el tipo, color y posición (coordenadas x, y de la matriz):

```
r n 1 1
t n 2 2
p b 4 2
r b 5 2
p n 0 3
p n 3 3
p b 0 4
p b 3 4
```

... y el siguiente fichero `main.cc`:

```
#include "ajedrez.h"

int main()
{
    Ajedrez j("Joaquim Traveset","ajedrez.txt");
    j.MoverPieza(5,2,5,1);
    j.Mostrar();
}
```

...el programa debería imprimir por pantalla:

```
Joaquim Traveset
| | | | | | | | |
| |r1| | | |r0| | |
| | |t1| |p0| | | |
|p1| | |p1| | | | |
|p0| | |p0| | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
```

Si en una casilla del tablero no se especifica ninguna pieza, se considerará que hay una pieza vacía (su tipo debe ser el caracter espacio). Las piezas blancas tendrán color=0, y las negras color=1.

El método `InicializarTablero` inicializa todas las piezas del tablero como vacías. El método `AnyadirPieza` debe añadir una pieza en las coordenadas x, y. Para ello, se debe comprobar que dicha casilla no contiene ya una pieza del mismo color (en este caso, mostrará un error y devolverá false).

Lo mismo ocurre en `MoverPieza`, no se puede mover una pieza a una casilla ocupada por otra del mismo color. El método `BorrarPieza` quita una pieza del tablero. Para ello, hay que comprobar que la casilla esté ocupada y, si no es así, mostrar un error y devolver false. Todos los métodos que reciban coordenadas por parámetro deben comprobar que estén entre 0 y 7 invocando al método `ComprobarCoordenadas`.

Ayuda: Puedes usar el siguiente `makefile` para compilar el programa:

```
main : main.o ajedrez.o pieza.o
      g++ -o main main.o ajedrez.o pieza.o
main.o: main.cc pieza.h ajedrez.h
      g++ -c -g -Wall main.cc
ajedrez.o: ajedrez.cc ajedrez.h pieza.h
      g++ -c -g -Wall ajedrez.cc
pieza.o: pieza.cc pieza.h
      g++ -c -g -Wall pieza.cc
```