



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



FUNDAMENTOS DE PROGRAMACIÓN EXAMEN PRÁCTICO 2

(Enero 2013)

!Importante! No se corregirá ninguna práctica que no cumpla escrupulosamente las normas que aparecen a continuación

Para la realización del examen

- Para iniciar sesión en las aulas de prácticas, tendrá que introducir su identificador de usuario y contraseña como siempre. En la casilla correspondiente a **Código: fpexamen**. Con este código sólo tendrá acceso Internet a la página de la asignatura. No tendrá disponible su unidad en red U : ni los puertos USB.
- Se creará un fichero fuente con la solución del problema, nombrándolo **problema.cpp**.
- Este archivo debe contener en la cabecera, como comentario, la siguiente información: nombre y apellidos, DNI y grupo de prácticas al que se asiste.
- En la evaluación de las prácticas se tendrá en cuenta, además de la corrección de la solución propuesta, el estilo de programación, el uso correcto de espacios y tabuladores, así como la claridad del código fuente.
- La entrega de la práctica se hará durante el tiempo de duración del examen. Para ello, se usará el sistema de gestión automática de prácticas (GAP) accesible a partir de la página web de la asignatura (Menú Principal → GAP).
- Debéis asegurarnos de entregar en vuestro grupo de prácticas: **13**. Se informa a los alumnos que el programa GAP contiene un módulo de detección de copias que será empleado durante la evaluación del examen. En caso de advertirse plagio se actuará tanto contra quien haya copiado como contra quien no haya velado por la privacidad de sus soluciones.
- La práctica se puede entregar tantas veces como se quiera durante el examen. El sistema GAP guarda la última entrega. De hecho, **se recomienda que se entregue varias veces a lo largo del examen**, ya que si el ordenador se quedara colgado, habría que reiniciarlo y se perdería toda la información.
- Tiempo de examen: **1.30 HORA**



Se desea analizar un texto y contar el número de veces que aparece cada letra. Para ello, debemos construir un programa que lea el texto desde la entrada estandar (cin) y que, al finalizar la ejecución, muestre en la salida estandar (cout) un listado con el número de veces que aparece cada letra del alfabeto en el texto original. Por ejemplo, si en la entrada escribimos el siguiente texto: Granada, enero de 2013. el programa mostraría el siguiente resultado:

A: 3, D: 2, E: 3, G: 1, N: 2, O: 1, R: 2 Observe que:

1. Tan solo se cuentan las letras del alfabeto.
2. No se distingue entre mayúsculas y minúsculas. Por ejemplo, la *a* se considera igual que la *A*.
3. Los símbolos que no sean una letra no se contabilizan (signos de puntuación, signos aritméticos, etc.). Tampoco se cuentan las letras con tilde, la ñ, etc.
4. La entrada de datos finaliza con el carácter '.'(punto).
5. El resultado muestra las letras en orden alfabético.

Restricciones del problema:

1. Debe implementar una clase *Frecuencias*. Dicha clase debe incorporar la funcionalidad necesaria para resolver el problema. La clase debe almacenar tanto las letras leídas como el número de repeticiones de cada una de ellas.
2. No se permite declarar vectores o cadenas fuera de la clase.

Ejemplo de fichero de validación:

En un lugar de la Mancha, de cuyo nombre .

Salida del programa: La frecuencia de cada letra de la entrada.

A: 4 B: 1 C: 2 D: 2 E: 4 G: 1 H: 1 L: 2 M: 2 N: 4 O: 2 R: 2 U: 3 Y: 1