DECSAI
Departamento de Ciencias de la Computación e I.A.

Fundamentos de Programación

Convocatoria de Septiembre. Curso 2011/2012

20 de Septiembre de 2012

1. (**3 puntos**) Se está desarrollando una aplicación para automatizar la realización de exámenes tipo test. El software incluirá una clase `Examen` que debe almacenar: el nombre de la asignatura, la lista de enunciados de las preguntas (cada enunciado es una cadena de caracteres) y la lista de respuestas correctas para cada pregunta (cada respuesta es un carácter). Implementa la clase junto con los siguientes métodos:

   - Un constructor que inicialice un objeto de tipo `Examen` dando el nombre de la asignatura y con la lista de preguntas vacía.
   - Un método `NuevaPregunta` que reciba un enunciado y la respuesta correcta y que los añada a la lista de preguntas del examen. Cada nueva pregunta siempre se añade al final de la lista.
   - Un método `NumPreguntas` que devuelva el número de preguntas de que consta el examen.
   - Un método `GetEnunciado` que devuelva el enunciado de la pregunta i-ésima.
   - Un método `GetRespuesta` que devuelva la respuesta de la pregunta i-ésima.

   Recuerda que los métodos deben ser robustos frente a entradas de datos erróneas.

   A continuación, se pide realizar un programa que permita evaluar a una serie de alumnos utilizando la clase `Examen`. El programa comenzará creando un objeto de tipo `Examen` y dándole contenido, es decir, leyendo las preguntas y respuestas correctas desde la entrada estándar y almacenándolas.

   Una vez leído el examen se procederá a la evaluación de un número de alumnos dado desde la entrada estándar. Para ello el programa le mostrará las preguntas del examen a cada alumno y leerá sus respuestas. Al finalizar cada alumno la prueba, el programa le dirá su nota de acuerdo a los siguientes criterios:

   - Por cada pregunta sin responder se suman 0 puntos.
   - Por cada respuesta correcta se suma 1 punto.
   - Por cada respuesta incorrecta se resta 1 punto.
   - La nota final estará en el intervalo [0, 10]. Un 10 significa que ha respondido y acertado todas las preguntas. Si la calificación es negativa se sustituye por cero.

   Observe que no es necesario almacenar las notas de los alumnos ya que se pueden ir mostrando al terminar cada uno de ellos la prueba. Además, podrá añadir nuevos métodos a la clase `Examen` si lo considera oportuno, así como implementar funciones externas a la misma. Se tendrá en cuenta la calidad del diseño propuesto.

2. (**2 puntos**) La ***criba de Eratóstenes*** (Cirene, 276 a. C. Alejandría, 194 a. C.) es un algoritmo que permite hallar todos los números primos menores que un número natural dado N.

   El procedimiento "manual" consiste en escribir todos los números naturales comprendidos entre 2 y N y *tachar* los números que *no* son primos de la siguiente manera: el primero (el 2) se declara primo y se tachan todos sus múltiplos; se busca el siguiente número entero que no ha sido tachado, se declara primo y se procede a tachar todos sus múltiplos, y así sucesivamente.

   Este proceso puede simplificarse. Considere que si c no es primo, entonces puede expresarse como el producto de dos números menores que él ($c = a \times b$). Por lo tanto, uno de ellos será menor o igual que $\sqrt{c}$.

   Escriba un programa que lea un número entero y muestre todos los primos menores que él empleando el algoritmo descrito.

3. (**1 punto**) Deseamos construir una función **recursiva** que muestre en la salida estándar los `m` mayores números pares que sean menores o iguales que un número `n`. El prototipo sería el siguiente:

   ```
   void MayoresPares(int m, int n);
   ```

   Por ejemplo:

   - `MayoresPares(3,6)` : Los 3 mayores números pares que son menores o iguales que 6 son estos: 6, 4, 2
   - `MayoresPares(4,3)` : Los 4 mayores números pares que son menores o iguales que 3 son estos: 2, 0, -2, -4
   - `MayoresPares(0,8)` : Los 0 mayores números pares que son menores o iguales que 8 son estos: (ninguno)