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

Fundamentos de Programación

Convocatoria de Febrero. Curso 2013/2014

31 de Febrero de 2014

Tiempo: 2h 30m

1. (**3 puntos**) Una matriz rectangular contiene valores enteros que representan medidas tomadas sobre una determinada región. Se supone que todos los valores deben ser positivos aunque por problemas de captación y registro algunos de ellos son negativos.

   Es preciso corregir estos valores erróneos y se propone sustituirlos por el valor promedio de sus **ocho** vecinos más cercanos espacialmente. Debe considerar que entre estos vecinos pudieran haber valores negativos, y en este caso **no** intervendrán en el cálculo del valor promedio:

   > Si hubiera un sólo valor negativo en la vecindad, se sumarán los valores de los 7 vecinos válidos y la suma se dividirá entre 7. Si hubiera dos valores negativos en la vecindad, se sumarán los valores de los 6 vecinos válidos y la suma se dividirá entre 6. ... Si no hubiera ningún valor válido, se sustituirá por un cero.

   Implemente un módulo para que dada una matriz rectangular como las descritas, devuelva **otra** matriz rectangular corregida. La matriz original **no** se modifica.

   Para la implementación debe considerar:

   a) El algoritmo debe ser simple y claro.

   b) Para simplificar, las casillas de los bordes **no** se modifican, aunque **sí** se usan para efectuar las correcciones oportunas. En definitiva, la primera y la última fila así como la primera y la última columna son iguales entre la matriz original y la corregida.

2. **(2.5 puntos)** Se dispone de un conjunto de cadenas de caracteres con los nombres de varios equipos de tenis de mesa participantes en un torneo. Se desea crear un vector que contenga todos los emparejamientos posibles. Por ejemplo, si se parte del conjunto { Albolote, Motril, Baza, La Zubia }, debe construirse el siguiente vector:

   {{Albolote, Motril} , {Albolote, Baza} , {Albolote, La Zubia} , {Motril, Baza} , {Motril, La Zubia} , {Baza, La Zubia}}

   Defina los tipos de datos y módulos necesarios para resolver el problema planteado. Tenga en cuenta que el número de equipos puede ser cualquiera (no sólo 4 como en el ejemplo), el orden de los emparejamientos no tiene por qué ser el anterior, sólo puede usar el tipo `string` para representar el nombre de cada equipo y no es necesario crear el programa principal.

3. **(3 puntos)** Existe un método para la clase `string` de C++, denominado `replace`, que cambia `n` caracteres de una cadena `cad1`, empezando en una determinada posición `pos`, por los caracteres presentes en una segunda cadena `cad2`. La llamada al método es `cad1.replace(pos, n, cad2)`. Ejemplos del funcionamiento de `replace` son:

```
string cad1="Fundamental Programación";
cad1.replace(9,2,"os de la");   // "al" -> "os de la"
                                // Ahora cad1 tiene "Fundamentos de la Programación"
cad1.replace(12,5,"en");        // "de la" -> "en"
                                // Ahora cad1 tiene "Fundamentos en Programación"
```

   Realizar un módulo llamado `reemplazar` con la misma funcionalidad que `replace`, para trabajar con cadenas de caracteres, pero con la restricción de que **no se puede** utilizar la clase `string`.

4. (**1.5 puntos**) Cree una función recursiva llamada `NumeroRectangulos` que determine cuántos rectángulos inscritos caben en un rectángulo, con la restricción de que sus áreas sean superior o igual a un valor dado. Todos los rectángulos inscritos comparten la esquina inferior izquierda, aunque las longitudes de sus lados se reducen a la mitad. Por ejemplo, dado el rectángulo de la figura la función devolverá 2 si el área mínima es 10. Observe que devolverá 3 si el área mínima fuera 2. Se deja al alumno la elección de los parámetros de la función con la salvedad de que **no se pasen longitudes de lados**. Se valorará el diseño correcto de la función. Indique la llamada a `NumeroRectangulos` desde `main`.