Sesión 13

► *Actividades a realizar en casa*

RELACIÓN extra.

1. (Examen Febrero 2015) En este ejercicio no hay que construir ninguna función o clase. Todo se programará en el main . Defina dos matrices de reales original y suavizada de tamaño 50 x 30 . Lea desde teclado los valores de la matriz original , obligando a que sea simétrica. Para ello, lea el número de filas n y a continuación introduzca los n x n datos de la matriz. Construya la matriz suavizada acorde a las siguientes indicaciones:

   a) La tabla resultante será simétrica.

   b) Los valores de la diagonal principal de la tabla resultante serán iguales a los de la tabla original.

   c) Los valores del triángulo superior de la tabla resultante se calculan de la siguiente manera: si (i, j) es una posición en el triángulo superior de la tabla resultante, su valor es el valor medio de los valores que ocupan las posiciones de las columnas $j, j+1, ..., n-1$ en la fila i de la tabla original.

2. (Examen Septiembre 2015) Buscaminas es un juego muy conocido cuyo objetivo es encontrar todas las minas existentes en un tablero rectangular, sin abrir ninguna. Si el jugador abre una mina, pierde la partida.

   Se pide definir la clase TableroBuscaMinas conteniendo lo siguiente:

   a) Para representar el tablero se trabajará con una matriz de datos 50x30 en la que todas las filas tienen el mismo número de columnas y los datos son de tipo bool. Contendrá un valor true en caso de haber una mina en la casilla especificada y false en caso contrario. Esta matriz será un dato miembro de la clase y al principio, todos los valores estarán a false.
   b) Un método para incluir una mina en una determinada casilla.
   c) Un método que reciba las coordenadas (i, j) de una casilla y devuelva un valor entero que indique el número de minas que rodean a la misma (será un número entre 0 y 8). En caso de que la casilla contenga una mina, se devolverá el valor -1 .

   Hay que tener especial cuidado con las casillas que hay en los bordes de la matriz ya que la casilla en la posición [0][0] , por ejemplo, sólo tiene tres vecinos rodeándola.

Incluya un pequeño programa de prueba para asignar algunas minas y obtener las minas que hay alrededor de algunas casillas.
Dificultad Baja.

3. (Examen Febrero 2009) Sgeún un etsduio de una uivenrsdiad ignlsea, no ipmotra el odren en el que las ltears etsan ersciats, la úicna csoa ipormtnate es que la pmrirea y la útlima ltera etsén ecsritas en la psioción cocrrtea. El rsteo peuden estar ttaolmntee mal y aún pordás lerelo sin pobrleams. Etso es pquore no lemeos cada ltera por sí msima snio la paalbra cmoo un tdoo. Diremos que dos palabras son similares si la primera letra de ambas palabras es igual, la última letra de ambas palabras también es igual y el resto de las letras son las mismas pero no están necesariamente en las mismas posiciones. De esta forma, las palabras ttalomntee y totalmente son similares. Declare en el main dos objetos de la clase SecuenciaCaracteres y asígneles algunos caracteres de prueba. Implemente en el main un algoritmo que compruebe si los dos objetos son similares según el criterio anterior. Si lo necesita, puede añadir los métodos que estime oportunos a la clase SecuenciaCaracteres.

   Si le sirve de ayuda, utilice como base la descripción del siguiente algoritmo:

   Usaremos una secuencia de letras ya procesadas
   ```
   Comparar las primeras y últimas letras de cada palabra
   Si son iguales:
   Recorrer el resto de letras:
   Si la letra no está en la secuencia de procesadas:
   Añadirla a procesadas
   Contar el número de apariciones de la letra en cada palabra
   Si el número de apariciones es distinto, no son similares
   ```
   Dificultad Media.

4. Se quiere construir un programa para realizar estadísticas sobre datos meteorológicos. Para ello, se dispone del registro de un conjunto de medidas tomadas en el aeropuerto de Granada a lo largo de los 3 últimos meses. Para simplificar el problema, supondremos que todos los meses tienen treinta días, por lo que el número de días sobre los que se tienen datos es 90. Las medidas vienen organizadas de la siguiente forma: temperatura a las 07h de la mañana, temperatura a las 13h (ambas en grados centígrados) y precipitaciones en mm, todas ellas enteras. Para facilitar la prueba del programa, se ha preparado un fichero de datos llamado meteo.csv , que puede descargarse desde decsai . Se pide hallar el valor medio, el máximo y el mínimo mensual de cada una de las tres medidas. También debe calcular el valor medio, valor máximo y el mínimo mensual de la amplitud de las temperaturas esto es, la diferencia de temperatura que hay entre las 13h y las 07h.

   Defina la clase Meteo que contendrá como dato miembro una matriz de enteros con los datos metereológicos. Defina un método para añadir los cuatro datos correspondientes a un día.

5. Juego de la vida

   El matemático John HortonConway en 1970 creó un juego de la vida basado en tres reglas sencillas: nacimiento, muerte y supervivencia. Este juego simula la evolución de células a través de diferentes generaciones.

   El juego se desarrolla sobre un tablero representado por una matriz cuadrada de 20 x 20. Cada elemento de la matriz representa un espacio que puede estar vacío u ocupado por una célula en una generación actual. Para evolucionar a la siguiente generación, a partir de una posición inicial, el juego evoluciona de acuerdo a las siguientes reglas:

   a) Si una casilla vacía del tablero está rodeada por:
      - tres células vecinas (tres casillas ocupadas) entonces se crea vida, agregándose una célula en esa casilla.
      - más de tres células permanecería muerta por superpoblación.

   b) Si una célula (casilla ocupada) tiene:
      - menos de 2 células vecinas (dos casillas ocupadas), muere por aislamiento y la casilla correspondiente del tablero queda vacía,
      - más de 3 células vecinas, muere por sofocación y también se libera la casilla del tablero que le correspondía.
      - exactamente dos o tres vecinas sobrevive una generación más en el juego.

   NOTAS:
   - Observar que una casilla dependiendo de donde se encuentre en el tablero tienes un número determinado de vecinos. Si se encuentra en una de las esquinas tiene 3 casillas vecinas, si se encuentra en los laterales del tablero tiene 5 casillas vecinas y si es interior tiene 8 casillas vecinas.

   - Para determinar la evolución de una casilla se tendrá en cuenta el estado actual de las casillas vecinas y no el estado que tendrán en la siguiente generación.

   Se pide implementar:

   a) Diseñar la clase que incluya los datos miembro necesarios

   b) Desarrollar un constructor que inicialice el tablero de manera aleatoria utilizando ... el generador de números restringiendo a 2 valores 0 y 1.

   c) Desarrollar un método para incluir una célula en la posición (i,j) del tablero de juego.

   d) Desarrollar un método que reciba las coordenadas (i, j) y devuelva un valor entero que indique el número de células vecinas que están vivas.

e) Desarrollar un método que evolucione al tablero de la siguiente generación dependiendo de los valores actuales de las casillas. Para ello hay que tener en cuenta las reglas del juego mencionadas anteriormente.

f) Desarrollar un método que muestre en pantalla el tablero de juego.

g) Desarrollar un programa principal donde se muestre la evolución del tablero durante 5 generaciones.

Actividad: Resolución de problemas.

Resuelve los ejercicios siguientes:

1 suavizada

2 Buscaminas

3 etsduio

4 meteo

Actividades de Ampliación

Juego de la vida

Mira el material de estudio tema 4. Y las soluciones de los ejercicios, colgadas