23/06/2017
versión 1.1

Algoritmos y Estructuras de Datos
Ejercicios tipo 1er Examen Parcial

Temas:

- Asignaciones
- Estructuras de control
- Subprogramas: Funciones
- Estructuras de datos: Arrays
- Algoritmos de inserción, búsqueda y ordenamiento

1) **Display.** Se debe agregar una funcionalidad que permita rotar caracteres de una pantalla monocromática. Cada caracter está conformado por una matriz cuadrada (16 filas x 16 columnas) de booleanos y, según el valor recibido por parámetro, la función rotará 90, 180 o 270 grados la matriz hacia la derecha. El prototipo de la función es:

```
void rotar(bool[][], short);
```

Se pide:
Implementar la función rotar() contemplando las 3 alternativas. En caso de no recibir ninguno de esos 3 valores, no realizar ningún cambio.

Ejemplo de rotación 90º:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Pruebas: Desde la función main() crear, invocar la función mostrar, rotar 90 grados y volver a mostrar la matriz.

2) **GPS.** Un dispositivo IoT (Internet de las Cosas[1]) de control de logística recibe los datos de geolocalización de las todas las unidades móviles. Este dispositivo almacena temporalmente en memoria todas las posiciones geográficas en un vector bidimensional de 100 filas por 2 columnas: latitudes y longitudes (en grados decimales[2]). Este buffer[3] debe permitir ingresar nuevas mediciones, buscar errores y ordenar ascendentemente para un siguiente procesamiento de datos. Una representación gráfica de una parte del vector:

LAT	LON
40.741895	-73.989308
41.712706	-72.599470
46.906045	-77.168255
40.738725	-73.492355
40.460322	-74.446794
00.000000	-74.766766

Se pide:
Implementar la función **agregar()** que recibe los siguientes parámetros: el array de posiciones geográficas, los dos nuevos datos a ingresar (LAT y LON) y la cantidad de elementos almacenados. Si no hay capacidad, deberá retornar el valor -1. También se debe implementar la función **validar()** para localizar y eliminar elementos con errores. Los errores son elementos con valor 00.00000. En caso de localizar errores, se deberá borrar la fila completa (desplazando todos los elementos que corresponda) y retornando un entero que informe la cantidad de filas eliminadas. Por último también se deberá implementar la función **ordenar()** que ordenará ascendentemente el array por latitud o longitud, según la palabra recibida por parámetro (“LAT” o “LON”).

Nota: El ordenamiento debe estar basado en alguno de los 3 algoritmos específicos (burbuja, inserción o Shell).

[1] http://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT
[2] https://www.gps-coordinates.net/
[3] https://en.wikipedia.org/wiki/Data_buffer