

**Algoritmos y Estructuras de Datos**

Ejercicios tipo 1<sup>er</sup> Examen Parcial **Temas:** Asignaciones. Estructuras de control. Subprogramas: Funciones. Estructuras de datos: Arrays. Algoritmos de inserción, búsqueda y ordenamiento

**1. Encuesta.** Se deben registrar los resultados de una encuesta a clientes de una librería. El encuestador carga la cantidad de libros que los clientes dicen leer durante un año y el sistema registra en un array de 20 elementos la respuesta según el siguiente criterio: Más de 10 libros Asiduo (A), Entre 3 y 10 libros Frecuente (F) y por debajo de 3 Ocasional (O). Siempre se trabaja de a 20 encuestas por informe.

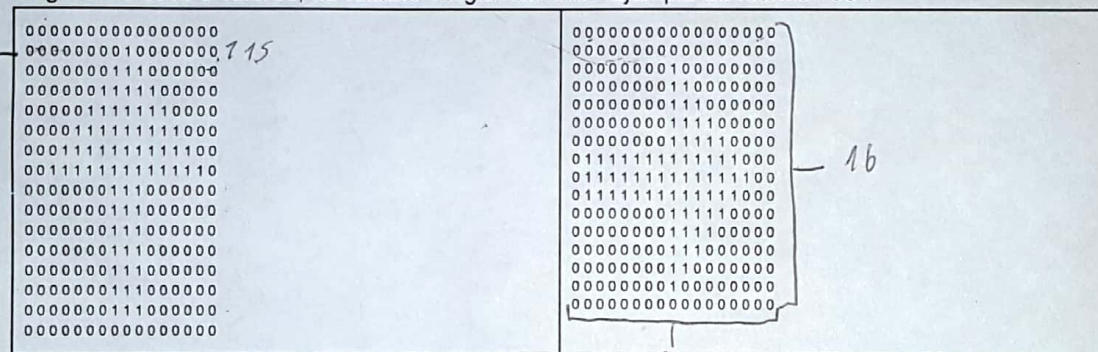
Se pide: Crear un programa que permita cargar la cantidad de libros leídos por cliente por pantalla y muestre, también por pantalla, cuántos se registraron de cada categoría. Modularizar mediante subprogramas para que desde una función principal se invoque a una específica para cargar datos y a otra para obtener el informe. *sin menú*

(25 puntos)

**2. 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:

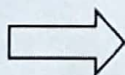
void rotar(bool[16][16], short);

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 90°:



(25 puntos)

**3. Código de barras.** Un sistema de gestión de logística debe almacenar en memoria los códigos de barra de los paquetes a medida que son leídos por un scanner. El líder técnico determinó que la mejor estructura para este caso sería una matriz bidimensional que almacene hasta 100 códigos. Como se desea que no haya repetición de datos, el array también almacenará la cantidad de lecturas sobre un mismo código, como se muestra en este ejemplo:



código	cantidad
01555668	2
39123439	1
01608798	4
0	1

Se pide: Crear la función **nuevaLectura()** que recibirá por parámetro: la matriz, el valor del índice al próximo elemento y el nuevo código a ingresar. Si el array no tiene más capacidad, deberá retornar el valor '-1'. También se deberá crear la definición de la función **validar()** que buscará lecturas erróneas (identificadas con el código de valor '0') y en cuyo caso se deberá eliminar la fila desplazando las siguientes de modo tal que no queden elementos vacíos intermedios (sólo podrá haber una fila con error). Por último, también se pide implementar la función **ordenar()** para ordenar las filas de la matriz ascendentemente por la columna 'cantidad'.

(50 puntos)



**Algoritmos y Estructuras de Datos. K1025. 1º Recuperatorio de 1º Examen Parcial.**

**Fecha: 23/08/2017**

**Apellido y nombre: ..... Legajo: .....**

Para aprobar debe sumar como mínimo 60 puntos (calif. 6), siendo 80 puntos el mínimo requerido para aprobación directa (calificación 8).

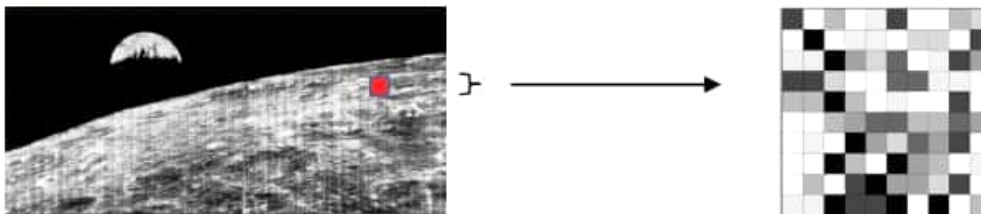
**1) Descuentos.** En una estructura de datos de tipo array de 20 elementos se almacena el importe de los artículos registrados por el cajero de un supermercado. Como existe un convenio entre el supermercado y un banco para otorgar un descuento del 10% a sus clientes, éste se debe aplicar sólo si el artículo no es parte del programa "precios cuidados" y si el cliente pagara con tarjeta de ese banco (ambos datos los indica el cajero).

Se pide: Crear un programa que permita cargar los importes de los productos por pantalla (siempre son 20) y muestre el resultado de calcular el total. Modularizar mediante subprogramas para que desde una función principal se invoque a una específica para cargar datos y a otra para mostrar el importe total por pantalla.

(25 puntos)

**2) Fotos.** Un sistema restaurador de imágenes debe realizar un preprocesamiento de la foto antes de iniciar la restauración. La imagen original (en escala de grises) se divide en matrices de 10 x 10 donde cada elemento puede tener un valor entre 0 y 255 siendo el valor 0 blanco (sin color) y 255 negro.

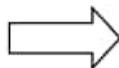
**Imagen:**



Se pide: Crear la función evaluarNitidez() que recibe como parámetro una matriz de enteros de 10 x 10 y que debe retornar el resultado de la suma de los valores de todos sus elementos.

(25 puntos)

**3) Mediciones.** Un conjunto fijo de 50 estaciones remotas meteorológicas reporta a intervalos fijos su medición de temperatura a un sistema central. Este sistema almacena las mediciones en una matriz de 50 x 2, registrando en la primera columna el identificador de la estación y el resultado de la medición en la segunda:



estación	medición
1556	22.5
8765	15.6
1112	-1
5569	25.3

Se pide: Implementar la función **nuevaMedicion()** que recibirá por parámetro: la matriz, su tamaño lógico (última fila con datos), el identificador de la estación y el valor de la medición (si ya existe la estación, se deberá actualizar el valor anterior). Otra funcionalidad requerida para implementar será la función **validar()** que buscará mediciones erróneas (identificadas con el valor '-1') y en cuyo caso se deberá eliminar la fila desplazando las siguientes de modo tal que no queden elementos vacíos intermedios. Por último, también se deberá implementar la función **buscar()** para buscar mediciones por identificador de estación y que retornará el valor de la medición asociada (-1 si no encuentra datos).

(50 puntos)



9 (nueve)

Algoritmos y Estructuras de Datos. 1° Examen Parcial.

Fecha:

Apellido y nombre:

Legajo:

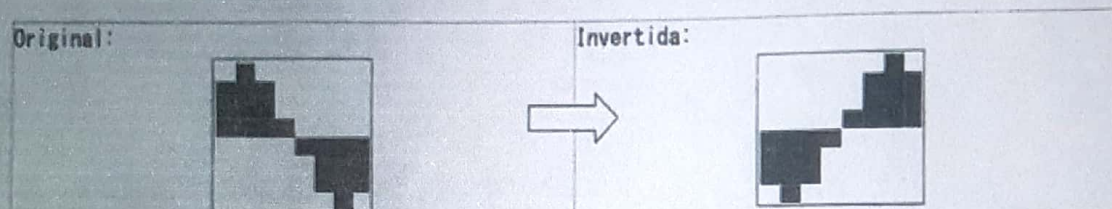
Para aprobar debe sumar como mínimo 60 puntos (calif. 6), siendo 80 puntos el mínimo requerido para aprobación directa (calificación 8).

- 26 1) **Calificaciones.** Una estructura de datos de tipo array de 30 elementos contiene las calificaciones (sin decimales) de todos los alumnos de un curso. Se desea contar con la posibilidad de obtener la mejor calificación, la menor y el promedio.

Se pide: Crear un programa que permita cargar las calificaciones por pantalla y muestre la calificación más alta, la menor y el promedio. Modularizar mediante subprogramas para que desde una función principal se invoque a una específica para cargar datos y a otra para obtener estadísticas.

(25 puntos)

- 23 2) **Audio.** Para ampliar un sistema de procesamiento de audio digital se desea agregar la funcionalidad de invertir el canal de entrada. La señal está representada por muestras de audio de 8 bits en matrices de 8 x 8 elementos, con la siguiente forma (original):



Se pide: Crear la función *invertir()* que recibe como parámetro un array bidimensional de booleanos de 8 x 8 y que lo modifica invirtiendo sus elementos en el plano vertical (invertida).

(25 puntos)

- 40 3) **Código de barras.** Un sistema de gestión de logística, debe almacenar en memoria los códigos de barra de los paquetes a medida que son leídos por un scanner. El líder técnico determinó que la mejor estructura para este caso sería una matriz bidimensional que almacene hasta 100 códigos. Como se desea que no haya repetición de datos, el array también almacenará la cantidad de lecturas sobre un mismo código, como se muestra en este ejemplo:



código	cantidad
01555668	2
39123439	1
01608798	4
0	1

Se pide: Implementar la función *nuevaLectura()* que recibirá por parámetro: la matriz, el valor del índice al último elemento cargado y el nuevo código a ingresar. Si el array no tiene más capacidad, deberá retornar el valor '-1'. Otra funcionalidad requerida para implementar será la función *validar()* que buscará lecturas erróneas (identificadas con el código de valor '0') y en cuyo caso se deberá eliminar la fila desplazando las siguientes de modo tal que no queden elementos vacíos intermedios (Nota: esta función se invocará luego de cada lectura, por lo que sólo podría haber 1 fila con error en todo el array). Por último, también se deberá implementar la función *ordenar()* para ordenar las filas de la matriz ascendentemente por la columna 'cantidad'.

(50 puntos)