

Apellidos.....

Nombre DNI

NOTAS IMPORTANTES.-

- Debes tener durante todo el examen tu dni o pasaporte sobre la mesa.
- Debes entrar en la cuenta que figura en tu examen (*login y password*). No debes salir de esta cuenta durante el examen y sólo tienes permitido utilizar un editor, el compilador y el depurador. Lee con mucha atención antes de comenzar a trabajar.
- Debes crear un fichero en tu cuenta cuyo nombre será: *loginuco.txt* (ej. *i92romeo.txt*) y contendrá tu nombre, apellidos y dni. En caso contrario no se corregirá tu examen.
- Debes realizar cada ejercicio en su directorio correspondiente. En caso contrario no se evaluará.
- Los ejercicios deberán compilar, en caso contrario, no se evaluarán.
- La puntuación indicada en los ejercicios es orientativa. El profesor podrá puntuar el ejercicio en conjunto para tener en cuenta la comprensión global de la asignatura.

TIEMPO: 4 horas

EJERCICIOS

(1 punto) Ejercicio 1 - Eliminatorio

Dos números amigos son dos enteros positivos a y b tales que a es igual a la suma de los divisores propios de b y b es igual a la suma de los divisores propios de a (la unidad se considera divisor propio, pero no lo es el mismo número). Escribe un programa que lea dos números enteros por teclado y determine si son amigos.

Ejemplo

El par de números (220,284)

- Los divisores propios de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110 que suman 284
- Los divisores propios de 284 son 1, 2, 4, 71 y 142 que suman 220

(2 puntos) Ejercicio 2 - Vectores

Un vector de n elementos se dice que es mayoritario si existe un elemento que aparece más de $n/2$ veces en él. Escribe un programa que usando funciones lea un vector de enteros, lo muestre por pantalla y realice las siguientes operaciones de manera secuencial:

- Determinar si el vector es mayoritario
- Borrar el elemento que ocupa la posición i -ésima, realizando los desplazamientos necesarios.

El programa deberá mostrar por pantalla de manera adecuada los resultados de las operaciones anteriores.

Ejemplo

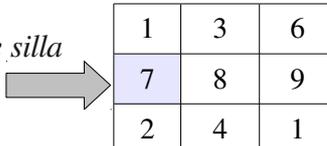
El vector (1, 2, 3, 1, 2, 1, 7, 1, 1, 1) es mayoritario ya que el número 1 aparece 6 veces en él.

Dado el vector (6, 8, 9, 1, 5) y la posición 2, el vector resultado será (6, 8, 1, 5)

(3 puntos) Ejercicio 3 - Matrices

Se dice que una matriz tiene un punto de silla si alguna posición de la matriz es el menor valor de su fila, y a la vez el mayor de su columna.

Punto de silla



1	3	6
7	8	9
2	4	1

Para calcular los puntos de silla de una matriz se necesitan la matriz original y una matriz auxiliar donde se marcarán los puntos de silla. El procedimiento para calcular los puntos de silla de una matriz es el siguiente:

1. Leer la matriz original
2. Inicializar la matriz auxiliar a 0
3. Calcular los mínimos de cada fila de la matriz original y sumar 1 a las posiciones que ocuparían los mínimos en la matriz auxiliar.
4. Calcular los máximos de cada columna de la matriz original y sumar 1 a las posiciones que ocuparían los máximos en la matriz auxiliar
5. Los elementos de la matriz auxiliar con valor igual a 2 son punto de silla de la matriz original.

0	0	0
0	0	0
0	0	0

1	0	0
1	0	0
0	0	1

1	0	0
2	1	1
0	0	1

(2) Inicializar matriz auxiliar

(3) Mínimos de cada fila

(4) Máximos de cada columna

Escribe un programa que calcule cuántos puntos de silla tiene una matriz. Para ello implementa las siguientes funciones sobre matrices estáticas de tipo entero (considera que el programa podrá trabajar como máximo con matrices de 20x20).

- **void rellenaMatriz(int matriz[][MAX], int nFil, int nCol).** Función que rellene una matriz con valores positivos leídos desde teclado. Los valores deberán estar en el intervalo [1-50].
- **void imprimeMatriz(int matriz[][MAX], int nFil, int nCol).** Función que imprime una matriz por pantalla.
- **void inicializaMatriz(int auxiliar[][MAX], int nFil, int nCol).** Función que inicialice todos los elementos de una matriz a 0.
- **void minimosFila(int datos[][MAX], int auxiliar[][MAX], int nFil, int nCol).** Función que calcula el mínimo de cada fila de la matriz *datos* y suma 1 a la casilla correspondiente de la matriz *auxiliar*.
- **void maximosColumna (int datos[][MAX], int auxiliar[][MAX],int nFil, int nCol).** Función que calcula el máximo de cada columna de la matriz *datos* y suma 1 a la posición correspondiente en la matriz *auxiliar*.
- **int puntosSilla(int auxiliar[][MAX], int nFil, int nCol).** Función que devuelva el número de puntos de silla que tiene una matriz (elementos con valor igual a 2).

Deberás estructurar tu programa en dos ficheros, *funciones.c* (contendrá las funciones descritas anteriormente) y *funciones.h* (contendrá los prototipos de las funciones). El fichero *main.c* que se encuentra en tu cuenta (en el directorio ejercicio3), contiene las llamadas a las funciones que has implementado; utilízalo para probar tu código.

(2 puntos) Ejercicio 4 - Cadenas

Escribe un programa que construya una cadena concatenando dos subcadenas. El programa leerá dos cadenas de caracteres y las posiciones de inicio y fin de las correspondientes subcadenas. Para ello debes implementar al menos las siguientes funciones:

- *extraeSubcadena*. Función genérica que reciba una cadena de caracteres y las posiciones de inicio y fin y extraiga la subcadena que hay entre inicio y fin.
- *concatena*. Función que reciba dos cadenas de caracteres y devuelva la concatenación de ambas en una nueva cadena.

Ejemplo

cadena1="Caperucita Roja" inicio subcadena1=0 fin subcadena1= 10 subcadena1="Caperucita "
cadena2="Ricitos de oro" inicio subcadena2=8 fin subcadena2= 13 subcadena2="de oro"

Cadena resultado de la concatenación de subcadena1 y subcadena2= "Caperucita de oro"

(2 puntos) Ejercicio 5 - Estructuras

Un punto en el espacio se puede representar mediante una estructura de dos campos (coordenada x y coordenada y). Escribe un programa que haciendo uso de esta estructura y utilizando funciones realice las siguientes operaciones con puntos

- Leer dos puntos a y b
- Mostrar los puntos por pantalla
- Calcular el punto medio de la recta que los une y mostrarlo por pantalla
- Calcular la distancia euclídea que existe entre los puntos a y b y mostrarla por pantalla

Ejemplo

Dados los puntos $a=(2, 6)$ y $b=(5, 3)$

Punto medio= $(3.5, 4.5)$

Distancia entre ambos puntos = $\sqrt{(2-5)^2+(6-3)^2}=4.2426$

Notas:

- Para que los ejercicios sean puntuados, han de funcionar correctamente y han de seguir las especificaciones reseñadas en cada uno de ellos.
- Excepto en las funciones específicas de entrada/salida y el *main*, no deben pedirse o mostrarse valores de variables dentro de las funciones. En caso contrario no se evaluará el ejercicio aunque funcione correctamente.
- La salida por pantalla será clara y ordenada, en caso contrario restarás un punto a tu nota final.
- Si los nombres de los archivos no se corresponden con los indicados en el enunciado del examen restarás un punto de tu nota final.
- Aquellos apartados que se implementen sin usar funciones no puntuarán, aunque funcionen correctamente.
- No se podrá hacer uso de variables globales.
- No olvides firmar tu examen y poner la hora de entrega.

Hora de entrega:

Firma: