stackoverflow
www.wuolah.com/student/stackoverflow

examenPracticoSeptiembre2015.pdf

Examenes Practicos MP

1° Metodología de la Programación

Grado en Ingeniería Informática

Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba

Apellidos......

Nombre ..... DNI .....

LOGIN: exmtp001 PASSWORD:osotwuca

NOTAS IMPORTANTES.-Lee con mucha atención antes de comenzar a trabajar.

- **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, el depurador y el man.**
- **Debes crear un fichero en tu cuenta de nombre *loginuco.txt* (ej. *i92romeo.txt*) con tus datos.**
- **Debes realizar cada ejercicio en su directorio correspondiente. En caso contrario no se evaluará.**
- **No se podrá salir del aula durante el examen, en caso contrario el examen se entenderá como entregado.**
- **La puntuación indicada en los ejercicios es orientativa.**
  - **El profesor o profesora podrá puntuar el ejercicio en conjunto para tener en cuenta la comprensión global de la asignatura.**
- **Los ejercicios deberán compilar. En caso contrario, no se evaluarán.**
- **La salida por pantalla será clara y ordenada; en caso contrario restarás un punto a tu nota final.**
- **Todos los ejercicios deben implementarse usando funciones.**
  - **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 de no cumplir estos requisitos, la nota final se verá penalizada.**
- **No se podrá hacer uso de variables globales.**
- **Los exámenes se pasarán por un software detector de copias.**
  - **Se considerarán copiados todos los alumnos implicados, con la consiguiente penalización.**

EJERCICIOS (Hay que usar funciones en todos) 2:30 horas

(3 puntos) Ejercicio 1

Implementa un programa que, **utilizando funciones**, realice las siguientes operaciones secuencialmente:

1. Crear una lista de ***N*** valores enteros.
2. Imprimir la lista por pantalla.
3. Determinar en una **única** función, el mayor y el menor valor de la lista, utilizando paso de parámetros por referencia.

Ten en cuenta lo siguiente:

- La función para mostrar la lista será una función recursiva.
- El mayor y el menor valor de la lista se imprimirá en desde el programa principal.

(2.5 puntos) Ejercicio 2

En el directorio *ejercicio2* de tu cuenta de examen, tienes un fichero binario (*productos.bin*) con información sobre productos que hay en una tienda. La información está almacena mediante la siguiente estructura:

```
struct producto {
   char nombre[50];
   int cod;
   float precio;
   int unidades;
};
```

Diseña y codifica una función que distribuya los productos del fichero binario en dos ficheros de texto. En un fichero se almacenarán los productos con más de X unidades y en el otro los productos con menos o igual número de unidades.

Ten en cuenta lo siguiente:

- El programa recibirá como argumentos en la línea de órdenes los nombres de los ficheros binario y texto.
- El programa deberá comprobar la existencia del fichero binario.
- El programa principal deberá preguntar el valor de X.
- Divide tu programa en 5 ficheros: *main.c texto.c binario.o texto.h* y *binario.h.*
- Utiliza inclusión condicional de código.
- En el directorio *ejercicio2* tienes un programa denominado *visualiza.exe* que te permite visualizar el fichero *productos.bin*.

(4.5 puntos) Ejercicio 3

En el directorio *ejercicio3* de tu cuenta de examen tienes tres ficheros que permiten trabajar con vectores de fracciones. Los ficheros son:

- *lee.o:* código objeto de la función para rellenar un vector de fracciones.
- *escribe.o*: código objeto de la función para imprimir por pantalla un vector de fracciones.
- *funciones.h*: prototipos de las funciones anteriores y estructura que define las fracciones.

Implementa un programa que, utilizando funciones, realice las siguientes operaciones secuencialmente:

1. Pedir por pantalla el valor de un número natural ***N***, que representará la longitud del vector.
   - Se pedirá este dato en la función *main*.
2. Reservar memoria para un vector dinámico de ***N*** elementos de tipo fraccion.
3. Rellenar el vector reservado.
4. Mostrar el vector por pantalla.
5. Ordenar el vector por cualquiera de los métodos vistos en clase.
   - Las fracciones se ordenarán en función del valor real que representan (numerado/denominador)
6. Mostrar el vector por pantalla.
7. Liberar memoria

Ten en cuenta lo siguiente:

- Utiliza las funciones proporcionadas para rellenar y visualizar el vector.
- El programa preguntará al usuario el sentido de la ordenación.
- La función de ordenación realizará la ordenación en uno u otro sentido mediante el uso de un **puntero a función** que se pasará como **parámetro**.
- Divide tu programa en 3 ficheros: *main.c vector.c vector.h.*
- Crea un fichero makefile con las siguientes características:
  - El ejecutable se creará a partir de *main.o*, *vector.o* y la librería *libVector.a*.
  - *La librería libVector.a se creará a partir de lee.o y escribe.o.*

Hora de entrega: **Firma:**

COMPETENCIAS EVALUADAS

CU2: *Conocer y perfeccionar el nivel de usuario en el ámbito de las TIC.*
CEB4: *Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.*