Una ONG nos ha encomendado el desarrollo de un sistema para organizar las donaciones que recibe, con el fin de identificar a sus donantes y las contribuciones que cada uno realiza.

Contamos con un archivo donde se describen los tipos de donaciones que la ONG recibe, el cual deberemos utilizar tal como se encuentra, con el fin de validar la carga de los nuevos donantes. Los tipos de donaciones que se reciben, son por ejemplo: Calzado, Juguetes, Ropa de Niños, Ropa de Adultos, Alimentos frescos, etc. Cada tipo de donación está identificada por un número entero, el cual se validará al momento de registrar un donante.

El archivo “tipoDonacion.dat” que se le ha compartido, contiene registros con la siguiente estructura:

typedef struct{

int idTipoDonacion;

char nombre[30];

int categoria; /// pueden ser 1, 2, 3 o 4

} stTipoDonacion;

La estructura de donante a utilizar es la siguiente:

typedef struct{

int idDonante;

char nombre[30];

float montoDonado;

int idTipoDonacion;

} stDonante;

La relación que existe entre las dos estructuras se da por el idTipoDonacion. Un tipo de donación puede tener muchos donantes. La categoría de la donación pueden ser 1, 2, 3 o 4.

Se nos solicita generar la siguiente funcionalidad:

1. Efectuar la carga nuevos donantes sobre un arreglo de una dimensión específica con intervención del usuario del sistema. Esta función deberá validar que el idTipoDonacion ingresado exista en el archivo de tipos de donación que se le ha entregado. Dicha operación se debe realizar en una función aparte (Modularizar).

Una vez cargado el arreglo de donantes, deberá mostrarlos por pantalla a fin de corroborar que los datos han sido debidamente ingresados. Modularizar.

1. A partir del arreglo cargado en el inciso anterior :
   1. Pasar a dos archivos de donantes (“donantesA.dat” y “donantesB.dat”) la información cargada en el arreglo, dependiendo del monto donado por cada uno. Se considera un donante “A” si ha contribuido con un monto superior a $25.000,00. Dentro de la función, deberá trabajar con dos variables de tipo FILE \*, una para cada archivo.
   2. Realizar una función que muestre un archivo de donantes por pantalla. Modularizar.
2. Averiguar el promedio del valor donado sobre el total de donantes almacenados en uno de los archivos. Para lo cual deberá realizar:
   1. Una función que calcule la cantidad total de registros que posee un archivo.
   2. Una función que calcule el promedio del valor donado sobre el total de donantes almacenados en el archivo.
   3. Un subprograma, que deberá invocar a las funciones anteriores (3.a y 3.b) e informar por pantalla los resultados obtenidos en cada caso. Deberá trabajar en el archivo correspondiente de donantes (donantes A o donantes B), de acuerdo a la solicitado por el usuario del sistema.
3. Mostrar todos los nombres de los tipos de donación de una categoría determinada. Para lo cual deberá realizar:
   1. Hacer una función que pase los nombres de los tipos de donación almacenados en el archivo a un arreglo de strings, correspondientes a una categoría específica. La función debe retornar la cantidad de nombres copiados.
   2. Hacer una función que muestre por pantalla un arreglo de strings cargado con la función anterior, indicando la categoría a la que corresponden.
   3. Un subprograma, que deberá invocar a las funciones anteriores (4.a y 4.b) e informar por pantalla los resultados obtenidos en cada caso.
4. Hacer una función main() que invoque a los modulos anteriores y demuestre el funcionamiento del programa.
   1. Para hacer esto, cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
   2. Muestre los resultados cada vez que sea necesario.
   3. Si lo considera, cree un menú de opciones para ejecutar cada función o subprograma.
   4. A fin de identificar cada inciso, comente su código indicando a qué apartado corresponde, por ejemplo: // Apartado 3.b

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Apartado | 1 | | | 2 | | 3 | | | 4 | | | 5 | |
| a | b | c | a | b | a | b | c | a | b | c | main | compilación |
| Puntaje | 10 | 10 | 7,5 | 12,5 | 7,5 | 7,5 | 7,5 | 7,5 | 5 | 5 | 5 | 10 | 5 |

**Tabla de puntuación:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Obtenido** | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| **Nota** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Condición** | Desaprobado | | | | | Aprobado | | | | |