

Algo1Mendez - Final - 17/09/2020

Ejercicio 1

Versión 1

Merry es un gran fanático de los juegos de mesa, aprovecha cada tiempito libre que tiene para jugar una partida de lo que sea, con quien sea. Tal es su fanatismo que lleva registro de cada partida de cada juego que jugó, como le fue y contra quién jugó en un archivo de texto llamado `partidas.csv`, el cual está ordenado ascendentemente por nombre del contrincante y luego por fecha. Cada línea del archivo contiene la siguiente información:

```
fecha;nombre del contrincante;id_juego;resultado de la partida
```

Se cuenta también un archivo binario de acceso directo ordenado por **id_juego**, llamado **juegos.dat**, que cuenta con todos los nombres de los juegos que Merry puede jugar.

En este archivo, los **id_juego** son números enteros correlativos que comienzan desde el 1 y tiene registros del tipo **juego_t**:

```
typedef struct juego {
    int id_juego;
    char nombre[MAX_NOMBRE];
} juego_t;
```

Se pide

1. Listar todas las partidas jugadas por Merry, agrupadas por contrincante, mostrando al finalizar cada uno, la cantidad de veces que ganó, perdió y empató, como se muestra a continuación:

Ejemplo

```
Bilbo
  Damas - Ganó.
  Backgammon - Perdió.
  Ajedrez - Perdió.
  ...
Ganó 15 veces, perdió 30, empató 10.
-----
Frodo
  Palitos chinos - Ganó.
  Batalla naval - Ganó.
  Palitos chinos - Perdió.
  ...
Ganó 43 veces, perdió 5, empató 18.
-----
```

Asumir todo lo que considere necesario para resolver el problema, como la convención para determinar si ganó, perdió o empató, etc.

Versión 2

Merry es un gran fanático de los juegos de mesa, aprovecha cada tiempito libre que tiene para jugar una partida de lo que sea, con quien sea. Tal es su fanatismo que lleva registro de cada partida de cada juego que jugó, como le fue y contra quién jugó en un archivo de texto llamado `partidas.csv`, el cual está ordenado ascendentemente por nombre del contrincante y luego por fecha. Cada línea del archivo contiene la siguiente información:

```
fecha;nombre del contrincante;id_juego;resultado de la partida
```

Se cuenta también un archivo binario de acceso secuencial, llamado **juegos.dat**, que cuenta con todos los nombres de los juegos que Merry puede jugar. Este archivo **no está ordenado**, pero se sabe que **no contiene más de 100 registros** del tipo **juego_t**:

```
typedef struct juego {
    int id_juego;
    char nombre[MAX_NOMBRE];
} juego_t;
```

Se pide

1. Listar todas las partidas jugadas por Merry, agrupadas por contrincante, y luego por fecha, mostrando al finalizar cada uno, la cantidad de veces jugó cada

día, como se muestra a continuación:

Ejemplo

Bilbo

20200110

Damas - Ganó.

Backgammon - Perdió.

Ajedrez - Perdió.

El 20200101 jugó 10 veces.

20200123

Quien es quien - Ganó.

Ajedrez - Ganó.

Truco - Perdió.

El 20200123 jugó 6 veces.

...

Frodo

20200101

Damas - Ganó.

Backgammon - Perdió.

Ajedrez - Perdió.

El 20200101 jugó 10 veces.

20200123

Quien es quien - Ganó.

Ajedrez - Ganó.

Truco - Perdió.

El 20200123 jugó 6 veces.

...

Asumir todo lo que considere necesario para resolver el problema, como la convención para determinar si ganó, perdió o empató, etc.

Ejercicio 2

Versión 1

Faramir dejó un testamento, porque sabía que, con la personalidad que tiene, algún día no volvería a casa.

Tuvo la precaución de encriptarlo, para que solo los herederos puedan descifrarlo ya que incluye indicaciones para llegar una fortuna oculta.

Siendo un merecido heredero de la herencia de Faramir se cuenta con el archivo de texto **testamento.txt** y se quiere desencriptar para acceder a la fortuna.

Se sabe que: - Las **a** fueron cambiadas por **s**. - Las **s** fueron cambiadas por **n**. - Las **n** fueron cambiadas por **r**. - Las **r** fueron cambiadas por **a**.

A su vez, por cada **e** que aparacia en el estamento, se agregó una risa socarrona **jeje**.

Se pide:

1. Crear un programa que **desencripte** el testamento de Faramir.

Ejemplo

Frase encriptada:

Ejejer ejejentejeje humildejeje scto, dejejeje min cejejertsvon dejejebsjo dejeje ls gasr T.

Frase desencriptada:

En este humilde acto, dejo mis centavos debajo de la gran T.

Versión 2

Faramir dejó un testamento, porque sabía que, con la personalidad que tiene, algún día no volvería a casa.

Quiere tomarse el trabajo de encriptarlo, para que solo los herederos puedan descifrarlo ya que incluye indicaciones para llegar una fortuna oculta.

Se tiene una version sin encriptar del testamento de Faramir en un archivo de texto llamado **testamento.txt** y se quiere encriptar para cumplir con lo pedido por él.

Se sabe que: - Las **a** deben ser cambiadas por **s**. - Las **s** deben ser cambiadas por **n**. - Las **n** deben ser cambiadas por **r**. - Las **r** deben ser cambiadas por **a**.

A su vez, por cada **e** que aparezca en el estamento, se debe agregar una risa socarrona **jeje**.

Se pide:

1. Crear un programa que **encripte** el testamento de Faramir.

Ejemplo

Frase desencriptada:

En este humilde acto, dejo mis centavos debajo de la gran T.

Frase encriptada:

Ejejer ejejentejeje humildejeje scto, dejejeje min cejejertsvon dejejebsjo dejeje ls gasr T.

Ejercicio 3

Versión 1

Dos armarios llenos de zapatillas tiene Sam, uno con las izquierdas y otro con las derechas, es como un toc que tiene que no le deja juntar los pares.

Algo que le gusta mucho, porque le copa estar a la moda, es cambiar los cordones para crear nuevas tendencias.

Un día se puso a ordenarlas y decidió crear un archivo binario de acceso secuencial por cada armario, los archivos están ordenados por código de zapatilla y tienen registros del tipo **zapatilla_t**.

```
typedef struct zapatilla {  
    int codigo;  
    char marca[MAX_MARCA];  
    int talle;  
    char color_cordon;  
    char armario;  
} zapatilla;
```

Dos zapatillas con el mismo código forman el par.

Se pide

1. Crear un archivo con aquellas zapatillas que no tienen su par en el otro armario.
2. Devolver, como retorno de la función, de las zapatillas que sí se puede completar el par, la cantidad que tienen cordones de diferente color entre ellas.

Versión 2

Dos armarios llenos de zapatillas tiene Sam, uno con las izquierdas y otro con las derechas, es como un toc que tiene que no le deja juntar los pares.

Algo que le gusta mucho, porque le copa estar a la moda es cambiar los cordones para crear nuevas tendencias.

Un día se puso a ordenarlas y decidió crear un archivo binario de acceso secuencial por cada armario, los archivos están ordenados por código de zapatilla y tienen registros del tipo **zapatilla_t**.

```
typedef struct zapatilla {  
    int codigo;  
    char marca[MAX_MARCA];  
    int talle;  
    char color_cordon;  
    char armario;  
} zapatilla;
```

Dos zapatillas con el mismo código forman el par.

Se pide

1. Crear un archivo con aquellas zapatillas que tienen su par en el otro armario, y guardar ambas.
2. Devolver, como retorno de la función, en cual de los armarios hay más zapatillas.

Ejercicio 4

Versión 1

Aburridos luego de haber salvado a la humanidad, Galadriel y Celeborn se pusieron a jugar al Quien Es Quien.

El juego consiste en una matriz, donde cada celda es un personaje con características como color de pelo, colos de ojos, etc.

Cada jugador elige un personaje y el objetivo es adivinar que personaje eligió el otro, para esto, deben hacerse preguntas que deben ser respondias con **Si** o **No** e ir descartando los personajes que no cumplan la respuesta.

Cada celda del tablero tiene registros del tipo **personaje_t**:

```
typedef struct rasgo {
    char descripcion[MAX_DESCRIPCION];
    bool tiene;
} rasgo_t;

typedef struct personaje {
    char nombre[MAX_NOMBRE];
    rasgo_t rasgos[MAX_RASGO];
    int tope_rasgos;
    bool descartado;
} personaje_t;
```

Se pide

1. Dada una matriz que representa el tablero de uno de los jugadores y un rasgo recibidos por parámetro, descartar todos los personajes que no cumplan con el rasgo recibido.
2. En caso de quedar solo un personaje sin descartar, mostrar por pantalla "Elegiste a **nombre_del_participante**".

Ejemplo

```
Celeborn pregunta: ¿Tiene ojos negros?
Galadriel responde: Si
```

El procedimiento deberá recibir un rasgo como el siguiente

```
Descripcion: Ojos negros.
Tiene: true.
```

Y deberá descartar a todos los personajes que dentro de sus rasgos **NO** tengan ojos negros.

Ejercicio 5

Versión 1

Pippin anda con ganas de darle un regalo a cada uno de sus amigos (cuando sean sus cumpleaños), pero se dió cuenta que acordarse de la fecha de 20 personas es un poco difícil, por lo que decidió anotarlos en algún lado.

Al intentar pensar dónde y cómo hacerlo, se vió incapacitado de elegir y plantear una solución.

1. Según las información brindada: ¿Dónde, cómo y por qué le recomendarías a Pippin que anote y busque los cumpleaños de sus compañeros?. Realizar una explicación de la/s funciones y estructuras que recomendaría.
2. Implementar, según la opción elegida, cómo haría para **GUARDAR** el cumpleaños de un amigo.

Aclare y justifique todas las suposiciones que tomó y cree necesarias para solución elegida.

Versión 2

Pippin anda con ganas de darle un regalo a cada uno de sus amigos (cuando sean sus cumpleaños), pero se dió cuenta que acordarse de la fecha de 20 personas es un poco difícil, por lo que decidió anotarlos en algún lado.

Al intentar pensar dónde y cómo hacerlo, se vió incapacitado de elegir y plantear una solución.

1. Según las información brindada: ¿Dónde, cómo y por qué le recomendarías a Pippin que anote y busque los cumpleaños de sus compañeros?. Realizar una explicación de la/s funciones y estructuras que recomendaría.
2. Implementar, según la opción elegida, cómo haría para **BUSCAR** el cumpleaños de un amigo.

Aclare y justifique todas las suposiciones que tomó y cree necesarias para solución elegida.