RECUPERATORIO PP\_LABORATORIO1

ALUMNA: AGUAYO, FIORELLA MARINA

LEGAJO: 113186

MATERIA: LABORATORIO 1

DIVISION: 1C

PROFESOR: GERMAN SCARAFILO

FECHA: 26/11/2022

b. DER

ID EQUIPO

DESCRIPCION

DIRECTOR TECNICO

EQUIPO

SE COMPONE

(1, 1)

(1, M)

TIENE

FIGURITA

ALBUM

(1, 1)

ID FIGURITA

NOMBRE

POSICION

ID EQUIPO

ALTURA

PESO

AÑO INGRESO

ID CLUB

ESTADO

ES DORADA

ESTADO EN ALBUM

IDS\_FIGURITAS

DORADAS

ESTADO

ISEMPY

SE COMPONE

CLUB

ID CLUB

NOMBRE

ENTRENADOR

AÑO DE FUNDACION

enlace con video explicativo

https://drive.google.com/file/d/1LvMnn2Ln8FnPE1W7YsKVwM2ZwtrQ6YRR/view?usp=sharing

**typedef** **struct**

{

**int** id;

**char** nombre[30]];

**char** entrenador[30];];

**int** anioFundacion;

}**eClub**;

c. El enunciado de los informes anexados.

1. Listado de figuritas pegadas. Se deberá mostrar el nombre del jugador, el equipo y la posición.

2. Listado de figuritas repetidas. Se deberá mostrar el nombre del jugador, el equipo y la cantidad de repeticiones en la pila.

3. Listado de figuritas doradas que están pegadas en el álbum.

4. Cuántos sobres tuvo que comprar el coleccionista para llenar el álbum. Tener en cuenta que una vez llenado el álbum puede seguir abriendo sobres.

5. Cuánto dinero lleva gastado el coleccionista, teniendo en cuenta que cada paquete cuesta $170.

6. Cuánto dinero gastó el coleccionista para completar el álbum, teniendo en cuenta que cada paquete cuesta $170.

7. Seleccionar un equipo y listar todas las figuritas que estén pegadas.

8. Listar figuritas pegadas por el club con mayor cantidad de figuritas

d. Prototipo de cada función del núcleo del programa con la documentación correspondiente.

*\*\*\* PERFIL ADMINISTRADOR \*\*\**

/\*\*

\* \fn int altaFiguritas(eFigurita[], int, int\*, eEquipo[], int, eClub[], int)

\* \brief Funcion que da de alta una figurita

\*

\* \param arrayFiguritas array de estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de figuritas

\* \param id puntero a entero que guarda la direccion de memoria del ID de la figurita

\* \param arrayEquipos array de estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param arrayClubs array de estructura eClubs

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **altaFiguritas**(eFigurita arrayFiguritas[], **int** tamFiguritas, **int**\* id, eEquipo arrayEquipos[], **int** tamEquipos, eClub arrayClubs[], **int** tamClubs);

/\*\*

\* \fn int modificarFiguritas(eFigurita[], int, eEquipo[], int, eClub[], int)

\* \brief Funcion que modifica una figurita

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de figuritas

\* \param arrayEquipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param clubs array a estructura eClubs

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **modificarFiguritas**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo arrayEquipos[], **int** tamEquipos, eClub clubs[], **int** tamClubs);

/\*\*

\* \fn int listarFiguritasCreadas(eFigurita[], int, eEquipo[], int)

\* \brief Funcion que lista las figuritas que se crearon

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de figuritas

\* \param arrayEquipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param clubs array a estructura eClubs

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **listarFiguritasCreadas**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo arrayEquipos[], **int** tamEquipos, eClub clubs[], **int** tamClubs);

*\*\*\* PERFIL COLECCIONISTA \*\*\**

/\*\*

\* \fn int comprarPaqueteDeFiguritasRandom(eFigurita[], int, eEquipo[], int, eAlbum\*, eFigurita[], int, int\*, int\*, int\*, int\*, int\*, eClub[], int)

\* \brief Funcion que permite al usuario comprar un paquete de figuritas y en base a eso calcular precios

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de figuritas

\* \param arrayEquipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param album puntero a estructura album

\* \param arrayRepetidas array a estructura eFigurita

\* \param tamRepetidas variable entera que guarda el tamaño de figuritas repetidas

\* \param cantidadRepetidas cantidad de figuritas repetidas

\* \param contadorSobres cantidad de sobres comprados en el momento

\* \param cantidadSobresTotal cantidad de sobres total

\* \param precioAcumulado precio que se gasta mediante se compra

\* \param precioAlbumCompleto precio gastado en sobres para completar el album

\* \param clubs array de estructura eClub

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **comprarPaqueteDeFiguritasRandom**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo arrayEquipos[], **int** tamEquipos, eAlbum\* album, eFigurita arrayRepetidas[], **int** tamRepetidas, **int**\* cantidadRepetidas, **int**\* contadorSobres, **int**\* cantidadSobresTotal, **int**\* precioAcumulado, **int**\* precioAlbumCompleto, eClub clubs[], **int** tamClubs);

/\*\*

\* \fn int intercambiarFiguritas(eFigurita[], int, eEquipo[], int, eFigurita[], int, int\*, eClub[], int)

\* \brief Funcion que permite cambiar figuritas entre el coleccionista y el administrador

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de figuritas

\* \param equipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param arrayRepetidas array a estructura eFigurita

\* \param tamRepetidas variable entera que guarda el tamaño de figuritas repetidas

\* \param flag bandera que vale 1 cuando se cuando se compro un paquete

\* \param clubs array de estructura eClub

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **intercambiarFiguritas**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo equipos[], **int** tamEquipos, eFigurita arrayRepetidas[], **int** tamRepetidas, **int**\* flag, eClub clubs[], **int** tamClubs);

*\*\*\* INFORMES \*\*\**

/\*\*

\* \fn int listadoFiguritasPegadas(eFigurita[], int, eEquipo[], int)

\* \brief Funcion que lista las figuritas pegadas

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de figuritas

\* \param arrayEquipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **listadoFiguritasPegadas**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo arrayEquipos[], **int** tamEquipos);

/\*\*

\* \fn int listarFiguritasRepetidas(eFigurita[], int, eEquipo[], int, int\*)

\* \brief Funcion que lista las figuritas repetidas y muestra la cantidad de repeticiones

\*

\* \param arrayRepetidas array a estructura eFigurita

\* \param tamRepetidas variable entera que guarda el tamaño de repetidas

\* \param arrayEquipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param cantidadRepeticiones puntero a entero qeu guarda la cantidad de repeticiones

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **listarFiguritasRepetidas**(eFigurita arrayRepetidas[], **int** tamRepetidas, eEquipo arrayEquipos[], **int** tamEquipos);

/\*\*

\* \fn int listadoFiguritasDoradasPegadas(eFigurita[], int, eEquipo[], int, eClub[], int)

\* \brief Funcion que lista las figuritas doradas en estado PEGADO

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de las figuritas

\* \param arrayEquipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param clubs array a estructura eClubs

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **listadoFiguritasDoradasPegadas**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo arrayEquipos[], **int** tamEquipos, eClub clubs[], **int** tamClubs);

/\*\*

\* \fn int listarFiguritasPegadasPorEquipo(eFigurita[], int, eEquipo[], int, eClub[], int)

\* \brief Funcion que lista las figuritas pegadas por equipo

\*

\* \param arrayFiguritas array a estructura eFigurita

\* \param tamFiguritas variable entera que guarda el tamaño de las figuritas

\* \param equipos array a estructura eEquipo

\* \param tamEquipos variable entera que guarda el tamaño de equipos

\* \param clubs array a estructura eClubs

\* \param tamClubs variable entera que guarda el tamaño de clubs

\* \return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **listarFiguritasPegadasPorEquipo**(eFigurita arrayFiguritas[], **int** tamFiguritas, eEquipo equipos[], **int** tamEquipos, eClub clubs[], **int** tamClubs);

/\*\*

\* @fn int mostrarClubesConMasApariciones(eFigurita[], int, eEquipo[], int, eClub[], int)

\* @brief Funcion que muestra los clubes que mas aparecen entre los pegados

\*

\* @param arrayFiguritas array a estructura eFigurita

\* @param tamFiguritas variable entera que guarda el tamaño de figuritas

\* @param equipos array a estructura eEquipo

\* @param tamEquipos variable entera que guarda el tamaño de equipos

\* @param clubs array de estructura eClub

\* @param tamClubs variable entera que guarda el tamaño de clubs

\* @return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **mostrarClubesConMasApariciones**(**eFigurita** arrayFiguritas[], **int** tamFiguritas, **eEquipo** equipos[], **int** tamEquipos, **eClub** clubs[], **int** tamClubs);

/\*\*

\* @fn int contadorDeClubs(eFigurita[], int, eClub[], int, int[])

\* @brief Funcion que cuenta la cantidad de figuritas que hay por club

\*

\* @param arrayFiguritas array a estructura eFigurita

\* @param tamFiguritas variable entera que guarda el tamaño de figuritas

\* @param clubs array de estructura eClub

\* @param tamClubs variable entera que guarda el tamaño de clubs

\* @param contadores array de enteros que guarda los valores aumentados de cada club

\* @return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **contadorDeClubs**(**eFigurita** arrayFiguritas[], **int** tamFiguritas, **eClub** arrayClubs[], **int** tamClubs, **int** contadores[]);

/\*\*

\* @fn int buscarMayorCantidadDeClubs(int[], int)

\* @brief Funcion que busca la mayor cantidad de clubs mediante comparaciones entre contadores

\*

\* @param contadores array de enteros que guarda los valores aumentados de cada club

\* @param tamClubs variable entera que guarda el tamaño de clubs

\* @return todoOk retorna -1 si esta mal y 0 si esta bien

\*/

**int** **buscarMayorCantidadDeClubs**(**int** contadores[], **int** tamClubs);

/\*\*

\* @fn void pegarFigurita(eAlbum\*, int, eFigurita[], int)

\* @brief Funcion que pega una figurita en el album

\*

\* @param album

\* @param indiceLibre

\* @param arrayFiguritas

\* @param indiceRandom

\*/

**void** **pegarFigurita**(**eAlbum** \*album, **int** indiceLibre, **eFigurita** arrayFiguritas[], **int** indiceRandom);

/\*\*

\* @fn void calcularPrecioTotal(eAlbum\*, int\*, int\*, int\*, int\*, int\*)

\* @brief Funcion que calcula los precios totales de la compra de sobres

\*

\* @param album variable de tipo eAlbum que guarda un album

\* @param flagSobres bandera que cambia de estado dependiendo si el album esta lleno o no

\* @param cantidadSobresTotal variable que guarda la cantidad de sobres que se compraron en total

\* @param contadorSobres variable que guarda la cantidad de sobres a medida que se van comprando

\* @param precioAlbumCompleto variable que guarda lo que se gasto para completar el album

\* @param precioAcumulado variable que guarda el precio que se va acumulando con la compra de sobres

\*/

**void** **calcularPrecioTotal**(**eAlbum** \*album, **int** \*flagSobres, **int** \*cantidadSobresTotal, **int** \*contadorSobres, **int** \*precioAlbumCompleto, **int** \*precioAcumulado);

}