Imagen que contiene Logotipo

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente con confianza media

INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

INGENIERIA EN SISTEMAS COMPUTACIONALES

ALGORITMOS Y ESTRUCTURAS DE DATOS

2CV3

EXAMEN 3er Parcial

APLICACIÓN DE LOS ARBOLES BINARIOS

PROFESOR: SANCHEZ GARCIA OCTAVIO

ALUMNO: CORTES BUENDIA MARTIN FRANCISCO

19/01/23

La Liga de GitHub es la siguiente:

El examen que nos tocará realizar será el siguiente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para este examen se me ocurrió hacer un árbol binario y hacer diferentes archivos “.txt” para así poder tomarlos como una base de datos, en este programa los ordene alfabéticamente gracias a los árboles binarios.

La base de datos que ocupe fue la de los siguientes jugadores:

|  |  |
| --- | --- |
| Valentina Mexico 2pts  Ximena Mexico 2pts  Michael Canada 2pts  Paul Canada 2pts  Robert Canada 2 pts  Harper EUA 2 pts  Liam EUA 2 pts  Sophia EUA 2 pts  Wiliam EUA 2 pts  Antonio Italia 2 pts  Giovanna Italia 2 pts  Paola Italia 2 pts  Paolo Italia 2 pts  Francis Francia 2 pts  Mbappe Francia 2 pts  Pierre Francia 2 pts  Jake UK 2 pts  Oliver UK 2 pts  Tommy UK 2 pts  Wally UK 2 pts  Maria Colombia 2 pts  Samuel Colombia 2 pts  Karim Francia 2 pts  Gael Mexico 10 pts  Jennifer Canada 10 pts  Charlotte EUA 10 pts | Andrea Italia 10 pts  Claude Francia 10 pts  Harry UK 10 pts  Ana Colombia 10 pts  Sofia Mexico 10 pts  Mariah Canada 10 pts  Sebastian Mexico 10 pts  Kane UK 10 pts  Mateo Mexico 5 pts  Jhon Canada 5 pts  Emma EUA 5 pts  Anna Italia 5 pts  Dominique Francia 5 pts  Jack UK 5 pts  Emiliano Colombia 5 pts  Jose Colombia 5 pts  Linda Canada 5 pts  Henry EUA 5 pts  Francesco Italia 5 pts  Jean Francia 5 pts  Marie Francia 5 pts  Lewis UK 5 pts  Luciana Colombia 5 pts  Natalia Mexico 5 pts  Jaden EUA 5 pts |

El código fue el siguiente:

#include <stdlib.h>

#include <stdio.h>

typedef struct

{

char nombre[20];

char pais[20];

int puntos;

} participante;

// en esta parte cree una structura en donde le dire que ocupare un nombre maximo 20 caracteres al igual que un pais y un puntos de tipo entero

typedef struct n

{

participante p;

struct n \*izquierda;

struct n \*derecha;

} Nodo;

//creamos el nodo y le damos esos punteros a izquierda y derecha

void insertar(participante, Nodo \*\*);

int Cadenita(char \*, char \*);

void Orden(Nodo \*);

void posiciones(Nodo \*);

void nacion(Nodo \*);

//las funciones que ocupare

void insertar(participante p, Nodo \*\*raiz)

{

Nodo \*nuevo = (Nodo \*)malloc(sizeof(Nodo));

nuevo->p = p;

nuevo->derecha = NULL;

nuevo->izquierda = NULL;

Nodo \*padre;

Nodo \*actual;

actual = \*raiz;

if(actual == NULL)

{

\*raiz = nuevo;

return;

}

while(1)

{

padre = actual;

if(Cadenita(p.nombre, actual->p.nombre) == 1)

{

actual = actual->derecha;

if(actual == NULL)

{

padre->derecha = nuevo;

return;

}

}

else

{

actual = actual->izquierda;

if(actual == NULL)

{

padre->izquierda = nuevo;

return;

}

}

}

}

// aqui implemnete todo el arbol binario agregando que se va ordenar alfabeticamente los nodos

int Cadenita(char \*x, char \*y)

{

int i = 0, j = 0;

while(x[i] != 0 && y[i] != 0){

if(x[i] < y[i])

return -1;

else if(x[i] > y[i])

return 1;

if(x[i] == 0 || y[i] == 0)

if(x[i] != 0)

return 1;

else if(y[i] != 0)

return -1;

i++;

}

return 0;

}

//esta funcion es la que esta encargada de ordenarlo alfabeticamnete para llegar al arbol

void Orden(Nodo \*raiz)

{

if(raiz!= NULL){

Orden(raiz->izquierda);

printf("%s, %s, +%d puntos por su medalla\n ",raiz->p.nombre,raiz->p.pais,raiz->p.puntos);

Orden(raiz->derecha);

}

}

// funcion de ordenamiento podemos notar que es un inOrder para que se note bien ese orden descendente

void posicion(Nodo \*raiz)

{

if(raiz!= NULL){

posicion(raiz->izquierda);

printf(" %s obtuvo en total %d puntos en las ESCOlimpiadas\n ",raiz->p.pais,raiz->p.puntos);

posicion(raiz->derecha);

}

}

//funcion de ordenamiento que ocupe para imprimir los puntos obtenidos por pais

void nacion(Nodo \*raiz)

{

if(raiz!= NULL){

nacion(raiz->izquierda);

printf(" Participante: %s Pais: %s \n ",raiz->p.nombre,raiz->p.pais);

nacion(raiz->derecha);

}

}

//funcion de ordenamiento que implemente para imprimir los participantes con su pais

int main()

{

//implemnete un menu para que fuese mas comodo el poder manipularlo

int input = 0;

printf("Las ESCOlimpiadas 20223\n");

printf("Recordemos que los valores de las medallas son:...\nMedalla de Oro 10pts\nMedalla de Plata 5pts\nMedalla de Bronce 2pts\n");

do {

printf("\nQue Quieres Saber De la Competencia - \n");

printf("\t1 : Lista de los 51 Jugadores \n"); //puse 51 participantes ya que el examen decia (por lo menos 50) entonces agregue uno mas por cualquier cosa

printf("\t2 : Participantes de Mexico\n"); //mostrara los participantes de ese pais

printf("\t3 : Participantes de Canada \n");//mostrara los participantes de ese pais

printf("\t4 : Participantes de EUA \n");//mostrara los participantes de ese pais

printf("\t5 : Participantes de Francia \n");//mostrara los participantes de ese pais

printf("\t6 : Participantes de Unite Kingdom (UK) \n");//mostrara los participantes de ese pais

printf("\t7 : Participantes de Colombia \n");//mostrara los participantes de ese pais

printf("\t8 : Participantes de Italia \n");//mostrara los participantes de ese pais

printf("\t9 : Quienes Tuvieron Medalla De Oro? \n");// participantes que obtuvieron oro

printf("\t10 : Quienes Tuvieron Medalla De Plata?\n");// participantes que obtuvieron plata

printf("\t11 : Quienes Tuvieron Medalla De Bronce? \n");// participantes que obtuvieron bronce

printf("\t12 : La Suma De Los Puntos Por Sus Medallas \n"); //total de puntos obtenidos

printf("\n\t0 : Salir\n");

printf("\nSeleccionaste > ");

scanf(" %i", &input);

switch (input){

case 0:{

exit(0);

}

case 1:{

FILE \*archivo = fopen("ListaDe51Participantes.txt", "rt");//implemento el archivo y le indico que lea

participante aux; // creamos un nodo auxiliar

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0) //puse que sea >0 ya que al leer el archivo .txt al llegar a una linea que no pueda leer dar -1 entonces saldra del rango

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los Participantes Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 2:{

FILE \*archivo = fopen("ParticipantesMexico.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Mexicanos Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 3:{

FILE \*archivo = fopen("ParticipantesCanada.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Canadienses Ordenados Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 4:{

FILE \*archivo = fopen("ParticipantesEUA.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Americanos Ordenados por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 5:{

FILE \*archivo = fopen("ParticipantesFrancia.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Franceses Ordenados Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 6:{

FILE \*archivo = fopen("ParticipantesUK.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Inlgeses Ordenados Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 7:{

FILE \*archivo = fopen("ParticipantesColombia.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Colombianos Ordenados Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 8:{

FILE \*archivo = fopen("ParticipantesItalia.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los participantes Italianos Ordenados Alfabeticamente por Arbol Binario son: \n\n");

nacion(raiz);

break;

}

case 9:{

FILE \*archivo = fopen("BaseDatosOro.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los ganadores de Oro Ordenados Alfabeticamente por Arbol Binario son: \n\n");

Orden(raiz);

break;

}

case 10:{

FILE \*archivo = fopen("BaseDatosPlata.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los ganadores de Plata Ordenados Alfabeticamente por Arbol Binario son: \n\n");

Orden(raiz);

break;

}

case 11:{

FILE \*archivo = fopen("BaseDatosBronce.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los ganadores de Bronce Ordenados Alfabeticamente por Arbol Binario son: \n\n");

Orden(raiz);

break;

}

case 12:{

FILE \*archivo = fopen("BaseDeDatosPosiciones.txt", "rt");

participante aux;

Nodo \*raiz = NULL;

while(fscanf(archivo, "%s %s %d", aux.nombre, aux.pais, &aux.puntos) > 0)

{

//printf(" %s, %s, %d\n",aux.nombre,aux.pais,aux.puntos);

insertar(aux, &raiz);

}

printf("Los Paises Quedaron de la siguiente manera \n\n");

posicion(raiz);

break;

}

default:{

break;

}

}

} while (input != 0);

}

Y así se muestra de la siguiente manera al compilar

Recordemos que este programa nos pide ordenar alfabéticamente a los que quedaron con medalla de oro, plata y bronce

Así como mostrar el total de participantes y el total de participantes de cada país

Y también la suma de los puntajes obtenido por cada pis que son un total de 7 países.

Al compilara este programa podemos ver que implemente un menú:

Texto

Descripción generada automáticamente

Al darle la opción 1 nos muestra la lista del total de los participantes :

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

Ordenados alfabéticamente gracias al árbol binario

Al darle la opción 2:

Texto

Descripción generada automáticamente

Opción 3:

Texto

Descripción generada automáticamente

Opción 4:

Texto

Descripción generada automáticamente

Opción 5:

Texto

Descripción generada automáticamente

Opción 6:

Texto

Descripción generada automáticamente

Opción 7:

Texto

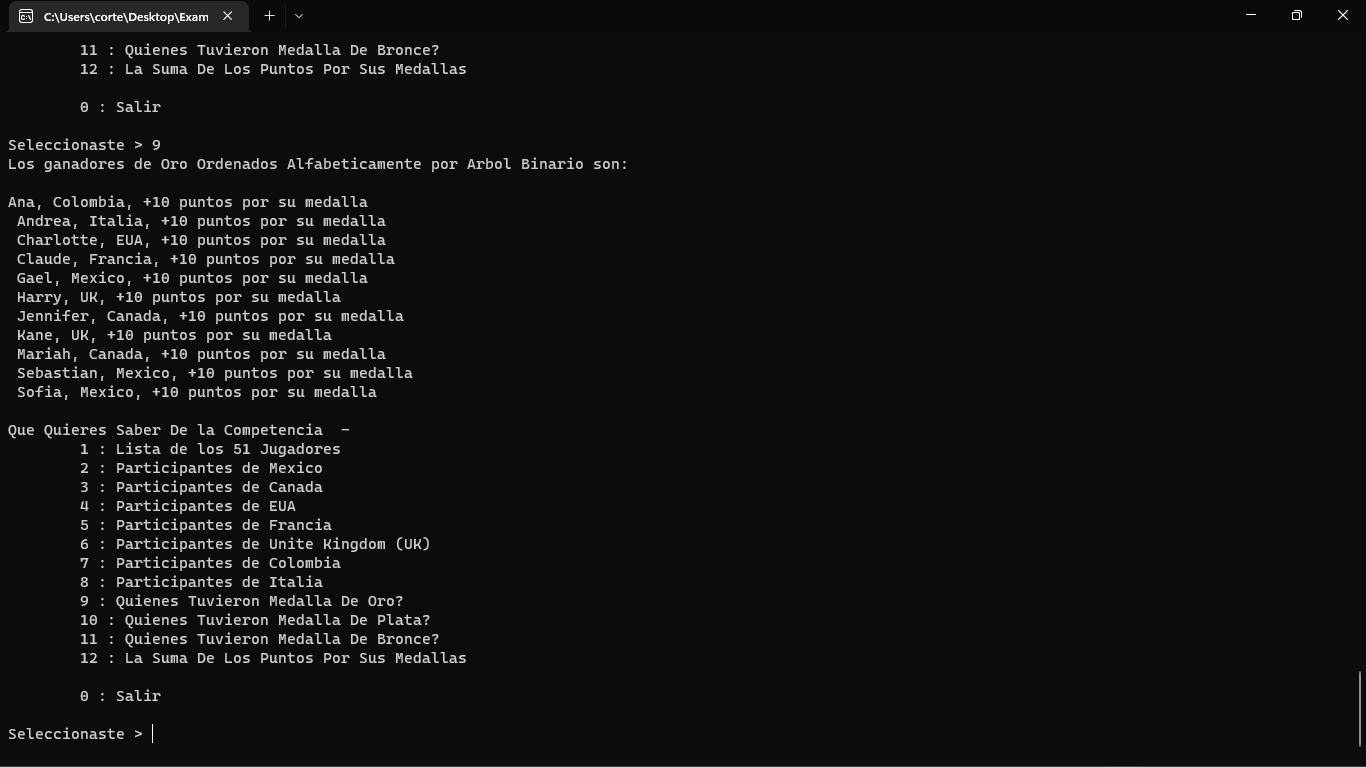
Descripción generada automáticamente

Opción 8:

Texto

Descripción generada automáticamente

Opción 9 (en esta opción vemos quienes obtuvieron oro y dieron 10 puntos a su nación):



Opción 10 (en esta opción vemos quienes obtuvieron plata y dieron 5 puntos a su nación):

Texto

Descripción generada automáticamente

Opción 11 (en esta opción vemos quienes obtuvieron bronce y dieron 2 puntos a su nación):

Texto

Descripción generada automáticamente

Y por ultimo el puntaje total de las naciones:

Texto

Descripción generada automáticamente