//NASIRIN, ABDUL M.

#include <iostream>

#include <fstream>//A LIBRARY FOR SAVING

using namespace std;

//CLASS PERSON

class Person {

// ATTRIBUTES

private:

bool IsEducated;

public:

int ID;

string Name;

int Age;

char Sex;

// CONSTRUCTOR

Person() {}

Person(int nID, string nName, int nAge, char nSex, bool nIsEducated){

ID = nID;

Name = nName;

Age = nAge;

Sex = nSex;

IsEducated = nIsEducated;

}

// ACCESSOR

bool GetIsEducated() {

return IsEducated;

}

//MUTATOR

void SetIsEducated(bool educated) {

IsEducated = educated;

}

// FUNCTIONS FOR DISPLAY

void DisplayPerson() {

cout << "ID: " << ID << endl;

cout << "Name: " << Name <<endl;

cout << "Age: " << Age <<endl;

cout << "Sex: " << Sex <<endl;

cout << "IsEducated: " << (IsEducated ? "Yes" : "No")<<endl;

}

};

//CLASS FOR MY ARRAY FUNCTIONS

class ArrayFunctions {

int def=0;

public:

// FUNCTION FOR MY INSERT

void InsertObject(Person p, Person PObj[10]) {

if(PObj[def].ID==0 && def<10) {

PObj[def] = p;

cout<<"Insert successful"<<endl;

def++;

system("cls");

}

else{

cout << "Insert unsuccessful" << endl;

}

}

// FUNCTION FOR MY DISPLAY

void DisplayArrayObjects(Person PObj[10]) {

cout<<"\_\_\_\_\_\_\_\_\_DISPLAY\_\_\_\_\_\_\_\_\_"<<endl;

for (int i=0; i<=def; i++) {

if (PObj[i].ID>0) {

PObj[i].DisplayPerson();

}

}

if (def==0) {

cout<< "Array is Empty";

}

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

}

// FUNCTION FOR MY SORTING

void SortArrayObjectsByID(Person PObj[10]){

for(int j=0; j<=def; j++) {

for(int i=j+1; i<=def; i++) {

if(PObj[i].ID<PObj[j].ID) {

Person temp=PObj[j];

PObj[j]=PObj[i];

PObj[i]=temp;

}

}

}

cout<< "Sorting successful" <<endl;

}

// FUNCTION FOR MY SEARCH

int SearchObjectByID(Person PObj[10], int searchid) {

for (int i=0; i<=def; i++) {

if (PObj[i].ID==searchid) {

return i;

}

}

cout << "ID Number " <<searchid << " Not Found."<< endl;

return -1;

}

void EditArrayObject(Person PObj[10], int searched) {

if (searched != -1) {

string newName;

cout << "Enter new Name: ";

cin >> newName;

PObj[searched].Name = newName;

char eduChoice;

cout << "Is the person educated? (y/n): ";

cin >> eduChoice;

bool newEducate = (eduChoice == 'y' || eduChoice == 'Y');

PObj[searched].SetIsEducated(newEducate);

cout << "Edit successful" << endl;

}

else {

cout << "Edit unsuccessful, object not found" << endl;

}

}

//FUNCTION FOR SAVING

void SaveObjectsToFile(Person PObj[10]) {

ofstream outputFile("objects.txt");

if (outputFile.is\_open()) {

for (int i = 0; i < 10; ++i) {

if (PObj[i].ID != 0) {

outputFile << " ID NUMBER : "<<PObj[i].ID << " || NAME : " << PObj[i].Name << " || AGE : " << PObj[i].Age << " || SEX : " << PObj[i].Sex << " || IS EDUCATED ?, 0 = NO , 1 = YES. : " << PObj[i].GetIsEducated() << endl;

}

}

cout << "Objects saved to file." << endl;

}

else {

cout << "Unable to open file to save objects." << endl;

}

outputFile.close();

}

};

int main() {

// array person

Person PObj[10];

// Class ArrayFunctions name

ArrayFunctions AFname;

// user choice for menu

int choice;

//do,While for the menu

do {

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_MENU\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout << "| (1.) Insert Object |"<<endl;

cout << "| (2.) Display Array Objects |"<<endl;

cout << "| (3.) Search Object by ID |"<<endl;

cout << "| (4.) Edit Array Object |"<<endl;

cout << "| (5.) Sort Array Objects by ID |"<<endl;

cout << "| (6.) Exit |"<<endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout << "Enter your number of your choice : ";

cin >> choice;

// user entererd choice

switch (choice) {

// case 1 is for run the insert function

case 1: {

int id, age;

string name;

char sex;

bool isEducated;

cout << "Enter ID Number : ";

cin >> id;

cout << "Enter Name : ";

cin >> name;

cout << "Enter Age : ";

cin >> age;

cout << "Enter Sex (M/F) : ";

cin >> sex;

cout << "IsEducated ? (1 for yes AND 0 for no) : ";

cin >> isEducated;

AFname.InsertObject(Person(id, name, age, sex, isEducated), PObj);

break;

}

// case 2 is for run the display function

case 2:{

AFname.DisplayArrayObjects(PObj);

system("pause");

system("cls");

break;

}

// case 3 is for run the searched function

case 3:{

int searchId;

cout << "Enter ID to search: ";

cin >> searchId;

int foundIndex = AFname.SearchObjectByID(PObj, searchId);

if (foundIndex != -1) {

cout << "Found ID Number " << " At Index: " << foundIndex << endl;

} else {

cout << "--Object not found--" << endl;

}

system("pause");

system("cls");

break;

}

// case 4 is for run the edit function

case 4: {

int searchId;

cout << "Enter ID to edit: ";

cin >> searchId;

int Index = AFname.SearchObjectByID(PObj, searchId);

AFname.EditArrayObject(PObj, Index);

system("pause");

system("cls");

break;

}

// case 5 is for run the sort function

case 5: {

AFname.SortArrayObjectsByID(PObj);

cout<<"--Try Display For The Result of Sorting--"<<endl;

system("pause");

system("cls");

break;

}

// case 6 is for terminate the program and save file

case 6:{

cout << "Exiting program." << endl;

AFname.SaveObjectsToFile(PObj);

break;

}

default:

cout << "Invalid choice. Please try again." << endl;

break;

}

}

while (choice != 6);

return 0;

}