#include <stdio.h>

#include <stdlib.h>

struct Student

{

char Name[64];

char ID[64];

int Age;

char Course[64];

double Grades[5][1];

};

int main()

{

int choice;

int count = 0;

do

{

system("color 02");

printf("\n<====== BSCS 2A Student Records ======>\n");

printf("> 1. Add student\n> 2. Search student\n> 3. NO. of student/s\n> 4. Records\n> 5. Delete Student\n> 6. Exit\nChoice: ");

scanf("%d", &choice);

struct Student student[100];

if(choice < 1 || choice > 6)

{

printf("Invalid choice.");

return;

}

switch(choice)

{

case 1:

Add(student, &count);

break;

case 2:

Search(student, &count);

break;

case 3:

printf("NO. of student/s: %d\n", Count(&count));

break;

case 4:

Record(student, &count);

break;

case 5:

Delete(student, &count);

break;

case 6:

printf("Exiting program........");

break;

default:

printf("Invalid input.");

}

}

while(choice != 6);

}

//Add

void Add(struct Student student[], int \*count)

{

int cmp;

struct Student newStudent;

getchar();

printf("Enter student name: ");

fgets(newStudent.Name, sizeof(newStudent.Name), stdin);

printf("Enter ID: ");

fgets(newStudent.ID, sizeof(newStudent.ID), stdin);

newStudent.Name[strcspn(newStudent.Name, "\n")] = '\0';

for(int i = 0; i < \*count; i++)

{

cmp = strcmp(student[i].ID, newStudent.ID);

if(cmp == 0)

{

printf("ERROR: ID number is already taken. Try another ID.");

return;

}

}

printf("Enter age: ");

scanf("%d", &newStudent.Age);

getchar();

if(newStudent.Age < 18 || newStudent.Age > 122)

{

printf("Invalid age.");

return;

}

printf("Enter student course: ");

fgets(newStudent.Course, sizeof(newStudent.Course), stdin);

printf("Enter grades:\n");

char \*subj[5] = {"Information Management", "Logic Design", "Programming Languages", "Software Engineering", "Ethics"};

printf("<--Subjects-->\n");

for(int i = 0; i < 5; i++)

{

printf("%s: ", subj[i]);

for(int j = 0; j < 1; j++)

{

scanf("%lf", &newStudent.Grades[i][j]);

if(newStudent.Grades[i][j] > 100)

{

printf("Invalid grade.");

return;

}

}

}

printf("> Student added successfully!");

student[\*count] = newStudent;

(\*count)++;

}

//Search

void Search(struct Student student[], int \*count)

{

char searchID[64];

int i, found = 0;

printf("Enter ID: ");

getchar();

fgets(searchID, sizeof(searchID), stdin);

for(i = 0; i < \*count; i++)

{

int cmp = strcmp(student[i].ID, searchID);

if(cmp == 0)

{

found = 1;

printf("Student ID: %s found!\n", searchID);

printf("<===Student's Information===>\n");

printf("Name: %s", student[i].Name);

printf("ID: %s\n", student[i].ID);

printf("Age: %d\n", student[i].Age);

printf("<==Grades==>\n");

GWA(student, count, i);

}

}

if(!found)

{

printf("Student ID: %s not found.", searchID);

}

}

//Records

void Record(struct Student student[], int \*count)

{

if(\*count == 0)

{

printf("Record is empty. Try adding a student.");

return;

}

else

{

printf("<===Records===>\n");

for(int i = 0; i < \*count; i++)

{

printf("Name: %s", student[i].Name);

printf("ID: %s", student[i].ID);

printf("<=====Grades=====>\n");

GWA(student, count, i);

printf("\n============================\n");

return;

}

}

}

//GWA

void GWA(struct Student student[], int \*count, int i)

{

double temp = 0.0;

for(int j = 0; j < 5; j++)

{

char \*subj[5] = {"Information Management", "Logic Design", "Programming Languages", "Software Engineering", "Ethics"};

for(int k = 0; k < 1; k++)

{

printf("%s: %.2lf\n", subj[j], student[i].Grades[j][k]);

temp += student[i].Grades[j][k];

}

}

printf("GWA: %.2lf", temp/5);

}

// Delete

void Delete(struct Student student[], int \*count)

{

char deleteID[64];

int i, cmp;

printf("Enter student ID to delete: ");

getchar();

fgets(deleteID, sizeof(deleteID), stdin);

for(i = 0; i < \*count; i++)

{

cmp = strcmp(student[i].ID, deleteID);

if(cmp == 0)

{

printf("Do you want to permanently delete student ID %s? [Y or N]?\nChoice: ", deleteID);

char choice;

scanf("%c", &choice);

if(choice == 'Y' || choice == 'y')

{

for(int j = i; j < \*count - 1; j++)

{

student[j] = student[j + 1];

}

(\*count)--; // Decrease the count

printf("Student with ID %s deleted successfully.\n", deleteID);

return;

}

else

{

printf("Deletion failed.");

return;

}

}

}

printf("Student not found with ID: %s.\n", deleteID);

}

//Count

int Count(int \*count)

{

return \*count;

}