#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//定义学生结构体

typedef struct student{

char name [100];

char uid [100] ;

}STU;

//题库结构体

typedef struct title{

char topic [100] ;

char a [100] ;

char b [100] ;

char c [100] ;

char d [100] ;

char answer [100] ;

}TIT;

// 函数声明

int UI();

void option(int n);

void student\_xinxi();

void kaoshi();

void shijuan();

void input();

void look();

void alter();

int main() {

int x = UI(); // x来接受UI返回值

option(x); //调用选择函数

// 打印成绩单

FILE \*fp\_score = fopen("成绩.txt", "r");

if (fp\_score!= NULL) {

char line[100]; //定义line数组来存放读取的内容

while (fgets(line, sizeof(line), fp\_score)) {

printf("%s", line);

}

fclose(fp\_score);

} else {

perror("无法打开成绩文件");

}

return 0;

}

// 考试系统UI设计

int UI() {

int t;

printf("考试管理系统\n");

printf("1、学生信息登录\n");

printf("2、进入考试\n");

printf("3、试卷系统录入\n");

scanf("%d", &t);

return t; // 通过返回值t来进行下一步的操作

}

// 开始界面判定操作，选择函数

void option(int n) {

switch (n) {

case 1:student\_xinxi();break;

case 2:kaoshi();break;

case 3:shijuan();break;

default:printf("管理员权限不够！请重新输入!\n");

int z;

scanf("%d", &z); //通过Z来判断case

option(z);

}

}

//考生信息系统

void student\_xinxi() {

STU stu1, stu2; //stu1存储用户输入的信息，stu2用于比对匹配

printf("姓名:\n");

scanf("%s", stu1.name );

printf("学号:\n");

scanf("%d", &stu1.uid );

FILE \*fp;

fp = fopen("考生信息表.txt", "r+"); //打开文件，进行读写操作

if (fp == NULL) { // 尝试创建文件

fp = fopen("考生信息表.txt", "w");

if (fp == NULL) {

printf("创建文件失败，无法进行录入操作！\n");

return ;

}

fwrite(&stu1, sizeof(STU), 1, fp); // 创建文件，开始录入操作

printf("考生信息已成功录入！\n");

fclose(fp);

} else {

fseek(fp, 0, SEEK\_END); // 将文件指针移动到文件末尾，获取文件大小

long int file\_size = ftell(fp); //通过file\_size的值是否为0判断是否为空文件

fseek(fp, 0, SEEK\_SET);

if (file\_size == 0) { //文件为空写入数据

fclose(fp);

fp = fopen("考生信息表.txt", "w");

if (fp == NULL) {

printf("重新打开文件失败(录入模式)\n");

return ;

}

fwrite(&stu1, sizeof(STU), 1, fp);

printf("考生信息已成功录入！\n");

fclose(fp);

} else {

int found = 0; //通过found值是否为0，匹配数据

while (fread(&stu2, sizeof(STU), 1, fp) == 1) {

if (strcmp(stu1.name, stu2.name ) == 0 && strcmp(stu1.uid ,stu2.uid ) == 0) {

found = 1;

break;

}

}

fclose(fp);

if (found == 1) {

printf("匹配成功！\n");

printf("是否开始考试！\n");

printf("YES(1) or NO(2)\n");

int YES; //通过YES的值来判断是否开始考试

scanf("%d",&YES);

if(YES == 1){

kaoshi();

} else {

printf("你已退出考试系统！\n");

UI(); //重新进入开始页面

}

} else {

printf("匹配失败！是否重新输入\n");

printf("T(1) or F(2)\n");

int TF; //通过s来判断是否要重新输入

scanf("%d",&TF);

if(TF == 1){

student\_xinxi();

} else {

printf("你已退出登录系统！\n");

UI(); //重新进入开始页面

}

}

}

}

}

// 试卷系统函数

void shijuan() {

printf("欢迎进入题库系统！\n");

printf("1、录入新题目\n");

printf("2、查看题库题目\n");

printf("3、修改题目\n");

int d; //通过d值来进行试卷系统操作

scanf("%d",&d);

switch (d){

case 1: input() ; break ;

case 2: look() ; break ;

case 3: alter() ; break ;

default :printf("管理员权限不够!请重新输入\n");

shijuan(); //重新进入试卷系统

}

}

// 考试函数

void kaoshi() {

FILE \*fp\_ques;

fp\_ques = fopen("试卷.txt", "r");

if (fp\_ques == NULL) {

perror("无法打开试卷文件");

return;

}

FILE \*fp\_score = fopen("成绩.txt", "a");

if (fp\_score == NULL) {

perror("无法打开成绩文件");

fclose(fp\_ques);

return;

}

char line[1024]; // 临时存储从试卷读取的内容

int score = 0; // 初始化成绩分数

int question\_count = 0; //初始化题目数量

while (fgets(line, sizeof(line), fp\_ques)) {

TIT ques; // 存储题目当前信息

line[strcspn(line, "\n")] = '\0';

strcpy(ques.topic, line);

fgets(line, sizeof(line), fp\_ques);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.a, line);

fgets(line, sizeof(line), fp\_ques);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.b, line);

fgets(line, sizeof(line), fp\_ques);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.c, line);

fgets(line, sizeof(line), fp\_ques);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.d, line);

fgets(line, sizeof(line), fp\_ques);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.answer, line);

// 输出题目

printf("题目：%s\n", ques.topic);

printf("A. %s\n", ques.a);

printf("B. %s\n", ques.b);

printf("C. %s\n", ques.c);

printf("D. %s\n", ques.d);

// 获取用户答案

char user\_answer[2]; // 用于存储考生输入的答案

printf("请输入你的答案：");

scanf("%s", user\_answer);

while (getchar()!= '\n');

if (user\_answer[0] == ques.answer[0]) {

score++;

}

question\_count++;

}

double average = (double)score / question\_count;

fprintf(fp\_score, "总分：%d，平均分：%.2f\n", score, average);

fclose(fp\_score);

fclose(fp\_ques);

printf("考试结束，成绩已保存！\n");

}

// 题目查看函数

void look() {

FILE \*fp = fopen("试卷.txt", "r");

if (!fp) {

perror("打开试卷文件失败");

return;

}

char line[1024]; // 临时存储文件内容

int question\_number = 1; // 题目序号初始化为1

while (fgets(line, sizeof(line), fp)) {

// 去除换行符

line[strcspn(line, "\n")] = '\0';

TIT temp; //文件内容复制到temp结构体变量中

strcpy(temp.topic, line);

if (fgets(line, sizeof(line), fp)) {

line[strcspn(line, "\n")] = '\0';

strcpy(temp.a, line);

}

if (fgets(line, sizeof(line), fp)) {

line[strcspn(line, "\n")] = '\0';

strcpy(temp.b, line);

}

if (fgets(line, sizeof(line), fp)) {

line[strcspn(line, "\n")] = '\0';

strcpy(temp.c, line);

}

if (fgets(line, sizeof(line), fp)) {

line[strcspn(line, "\n")] = '\0';

strcpy(temp.d, line);

}

if (fgets(line, sizeof(line), fp)) {

line[strcspn(line, "\n")] = '\0';

strcpy(temp.answer, line);

}

// 输出题目内容

printf("第 %d 题：\n", question\_number);

printf("题目：%s\n", temp.topic);

printf("A. %s\n", temp.a);

printf("B. %s\n", temp.b);

printf("C. %s\n", temp.c);

printf("D. %s\n", temp.d);

printf("答案：%s\n\n", temp.answer);

question\_number++;

}

fclose(fp);

}

// 题目输入函数

void input() {

FILE \*fp = fopen("试卷.txt", "a");

if (!fp) {

perror("无法打开试卷文件");

return;

}

int num; // num为录入的题目数量

printf("请输入要录入的题目数量：");

if (scanf("%d", &num)!= 1) {

fprintf(stderr, "输入题目数量格式错误\n");

fclose(fp);

return;

}

// 清除多余字符

while (getchar()!= '\n');

TIT ques; // ques存储录入的数据

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

printf("请输入第 %d 题题目：\n", i + 1);

fgets(ques.topic, sizeof(ques.topic), stdin);

ques.topic[strcspn(ques.topic, "\n")] = '\0';

printf("请输入 A 选项：\n");

fgets(ques.a, sizeof(ques.a), stdin);

ques.a[strcspn(ques.a, "\n")] = '\0';

printf("请输入 B 选项：\n");

fgets(ques.b, sizeof(ques.b), stdin);

ques.b[strcspn(ques.b, "\n")] = '\0';

printf("请输入 C 选项：\n");

fgets(ques.c, sizeof(ques.c), stdin);

ques.c[strcspn(ques.c, "\n")] = '\0';

printf("请输入 D 选项：\n");

fgets(ques.d, sizeof(ques.d), stdin);

ques.d[strcspn(ques.d, "\n")] = '\0';

printf("请输入正确答案：\n");

while (1) {

ques.answer[0] = getchar();

if (ques.answer[0] >= 'A' && ques.answer[0] <= 'D') {

ques.answer[1] = '\0';

getchar(); // 清除换行符

break;

}

printf("请输入大写字母 A - D：");

}

// 将当前题目完整的信息（题目、各选项、答案）按行格式写入到打开的试卷文件中

fprintf(fp, "%s\n%s\n%s\n%s\n%s\n%s\n", ques.topic, ques.a, ques.b, ques.c, ques.d, ques.answer);

}

fclose(fp);

printf("题目录入成功！\n");

}

// 题目修改函数

void alter () {

FILE \*fp = fopen("试卷.txt", "r+");

if (fp == NULL) {

perror("无法打开试卷文件");

return;

}

char line[1024];

int question\_number = 1;

int target\_question; // 定义要修改的题目序号

printf("请输入要修改的题目序号：");

scanf("%d", &target\_question);

while (getchar()!= '\n');

// 定位到目标题目

int current\_question = 1;

while (fgets(line, sizeof(line), fp)) {

if (current\_question == target\_question) {

TIT ques; // 临时存储读取内容

line[strcspn(line, "\n")] = '\0';

strcpy(ques.topic, line);

fgets(line, sizeof(line), fp);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.a, line);

fgets(line, sizeof(line), fp);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.b, line);

fgets(line, sizeof(line), fp);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.c, line);

fgets(line, sizeof(line), fp);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.d, line);

fgets(line, sizeof(line), fp);

line[strcspn(line, "\n")] = '\0';

strcpy(ques.answer, line);

// 输出原题目内容

printf("原题目：\n");

printf("题目：%s\n", ques.topic);

printf("A. %s\n", ques.a);

printf("B. %s\n", ques.b);

printf("C. %s\n", ques.c);

printf("D. %s\n", ques.d);

printf("答案：%s\n\n", ques.answer);

// 获取新内容

printf("请输入新题目：\n");

fgets(ques.topic, sizeof(ques.topic), stdin);

ques.topic[strcspn(ques.topic, "\n")] = '\0';

printf("请输入 A 选项：\n");

fgets(ques.a, sizeof(ques.a), stdin);

ques.a[strcspn(ques.a, "\n")] = '\0';

printf("请输入 B 选项：\n");

fgets(ques.b, sizeof(ques.b), stdin);

ques.b[strcspn(ques.b, "\n")] = '\0';

printf("请输入 C 选项：\n");

fgets(ques.c, sizeof(ques.c), stdin);

ques.c[strcspn(ques.c, "\n")] = '\0';

printf("请输入 D 选项：\n");

fgets(ques.d, sizeof(ques.d), stdin);

ques.d[strcspn(ques.d, "\n")] = '\0';

printf("请输入正确答案：\n");

while (1) {

ques.answer[0] = getchar();

if (ques.answer[0] >= 'A' && ques.answer[0] <= 'D') {

ques.answer[1] = '\0';

getchar(); // 清除换行符

break;

}

printf("请输入大写字母 A - D：");

}

// 将文件指针回退到当前题目的起始位置

// 计算需要回退的字节数，是题目、各选项、答案的长度总和再加6个换行符的长度

fseek(fp, - (strlen(ques.topic) + strlen(ques.a) + strlen(ques.b) + strlen(ques.c) + strlen(ques.d) + strlen(ques.answer) + 6), SEEK\_CUR);

// 将新的题目内容写入文件，覆盖原内容

fprintf(fp, "%s\n%s\n%s\n%s\n%s\n%s\n", ques.topic, ques.a, ques.b, ques.c, ques.d, ques.answer);

break;

}

// 计算当前题目已经读取的行数，用于定位下一题

int line\_count = 0;

while (fgets(line, sizeof(line), fp) && line\_count < 6) {

line\_count++;

}

current\_question++;

}

fclose(fp);

printf("题目修改成功！\n");

}