#include "stdafx.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define False 0

#define True 1

struct stu \*Adding(struct stu \*head);

struct stu \*Delete(struct stu \*head);

void Show(struct stu \*head);

void Classify(struct stu \*head);

void Find(struct stu \*head);

void Correct(struct stu \*head);

void Free(struct stu \*head);

void Sort(struct stu \*head);

void Save(struct stu \*head);

//学生信息的类别

struct stu

{

int num;

char name[10];

char sex[10];

char major[10];

struct stu \*next;

};

int main()

{

int judge = False;

int input;

int password;

int right\_password = 123;

char ID[10];

char right\_ID[10] = "luffy";

struct stu \*head = NULL;

printf("学生信息管理系统\n");

while (1)

{

printf("User name:");

scanf\_s("%s", ID, 10);

printf("Password:");

scanf\_s("%d", &password);

//验证所输入的账号密码是否正确

if (strcmp(ID, right\_ID) == 0 && password == right\_password)

{

judge = True;

printf("登录成功!\n\n");

break;

}

else

printf("ERROR\n\n");

}

if (judge == True)

{

while (1)

{

printf("\n欢迎来到学生信息管理系统\n");

printf("(1 录入信息\t");//学号姓名性别专业

printf("(2 信息删除\n");//学号

printf("(3 遍历信息\t");//学号排序

printf("(4 分类查看\n");//专业 性别

printf("(5 信息查询\t");//学号 姓名

printf("(6 信息修改\n");//学号

printf("(7 排序保存\t");

printf("(8 退出系统\n");

printf("请输入你想要执行的命令:");

scanf\_s("%d", &input);

switch (input)

{

case 1: head = Adding(head); break;

case 2: head = Delete(head); break;

case 3: Show(head); break;

case 4: Classify(head); break;

case 5: Find(head); break;

case 6: Correct(head); break;

case 7:

{

Sort(head);

Save(head);

break;

}

case 8: exit(0);

}

}

Free(head);

return 0;

}

}

//录入学生信息

struct stu \*Adding(struct stu \*head)

{

//建立链表并获取链表存储的内存

struct stu \*p = NULL, \*pr = head;

p = (struct stu \*)malloc(sizeof(struct stu));

//如果链表为空就退出

if (p == NULL)

exit(0);

//如果头指针为空 头指针为p

if (head == NULL)

head = p;

//如果头指针非空 末指针为pr p指向新的节点

else

{

while (pr->next != NULL)

pr = pr->next;

pr->next = p;

}

//获取学生信息

printf("请输入学号:");

scanf\_s("%d", &p->num);

printf("请输入学生姓名:");

scanf\_s("%s", p->name, 10);

printf("请输入性别:");

scanf\_s("%s", p->sex, 10);

printf("请输入专业:");

scanf\_s("%s", p->major, 10);

//改变末指针的指向

p->next = NULL;

printf("成功录入\n");

return head;

}

//学生信息排序

void Sort(struct stu \*head)

{

struct stu \*p1;

struct stu \*p2;

//以下四行为临时变量的定义

struct stu temp;

//使用冒泡排序法

for (p1 = head; p1 != NULL; p1 = p1->next)

for (p2 = p1->next; p2 != NULL; p2 = p2->next)

if (p1->num > p2->num)

{

temp.num = p1->num;

strcpy\_s(temp.name, 10, p1->name);

strcpy\_s(temp.sex, 10, p1->sex);

strcpy\_s(temp.major, 10, p1->major);

p1->num = p2->num;

strcpy\_s(p1->name, 10, p2->name);

strcpy\_s(p1->sex, 10, p2->sex);

strcpy\_s(p1->major, 10, p2->major);

p2->num = temp.num;

strcpy\_s(p2->name, 10, temp.name);

strcpy\_s(p2->sex, 10, temp.sex);

strcpy\_s(p2->major, 10, temp.major);

}

printf("排序成功\n");

}

//保存文件

void Save(struct stu \*head)

{

struct stu \*p = head;

FILE \*fp;

//打开文件并判断文件或头指针是否为空，如果为空就退出

fopen\_s(&fp, "file", "w");

if (fp == NULL || head == NULL)

exit(0);

//依次写入学生信息

while (p != NULL)

{

fprintf(fp, "num:%d\n", p->num);

fprintf(fp, "name:%s\n", p->name);

fprintf(fp, "sex:%s\n", p->sex);

fprintf(fp, "major:%s\n\n", p->major);

p = p->next;

}

//关闭文件

fclose(fp);

printf("已成功将学生信息写入%s\n", "file");

}

//删除学生信息

struct stu \*Delete(struct stu \*head)

{

struct stu \*p = head, \*pr = head;

int num;

printf("请输入你要删除的学生序号:");

scanf\_s("%d", &num);

if (head == NULL)

return(head);

//通过遍历寻找要删除的学号是否存在

while (num != p->num && p->next != NULL)

{

pr = p;

p = p->next;

}

//pr是p之前的一个指针

if (num == p->num)

{

if (p == head)

head = p->next;

else

pr->next = p->next;//跳过p指针指向的节点

printf("已删除该学生信息\n");

free(p);

}

else

printf("未查找到此学生信息\n");

return head;

}

//显示所有学生信息

void Show(struct stu \*head)

{

struct stu \*p = head;

if (p == NULL)

exit(0);

while (p != NULL)

{

printf("num:%d\n", p->num);

printf("name:%s\n", p->name);

printf("sex:%s\n", p->sex);

printf("major:%s\n\n", p->major);

p = p->next;

}

}

//学生信息分类查看

void Classify(struct stu \*head)

{

struct stu \*p = head;

int choice;

char major\_[10];

char sex\_[10];

if (p == NULL)

exit(0);

printf("(1 按专业输出 \t (2 按性别输出\n");

scanf\_s("%d", &choice);

if (choice == 1)

{

printf("请输入你想要查看的专业:");

scanf\_s("%s", major\_, 10);

//遍历链表，找出同专业的节点

while (p != NULL)

{

if (strcmp(major\_, p->major) == 0)

{

printf("num:%d\n", p->num);

printf("name:%s\n", p->name);

printf("sex:%s\n", p->sex);

printf("major:%s\n\n", p->major);

}

p = p->next;

}

}

if (choice == 2)

{

printf("请输入你想要查看的性别:");

scanf\_s("%s", sex\_, 10);

//遍历链表，找出同性别的节点

while (p != NULL)

{

if (strcmp(sex\_, p->sex) == 0)

{

printf("num:%d\n", p->num);

printf("name:%s\n", p->name);

printf("sex:%s\n", p->sex);

printf("major:%s\n\n", p->major);

}

p = p->next;

}

}

}

//查询学生信息

void Find(struct stu \*head)

{

struct stu \*p = head;

int num, choice;

char name[10];

printf("1) 按学号查找 \t 2) 按姓名查找\n");

scanf\_s("%d", &choice);

if (p == NULL)

exit(0);

if (choice == 1)

{

printf("请输入你要查找学生的学号:");

scanf\_s("%d", &num);

//遍历链表试图找出学生

while (p->next != NULL && num != p->num)

p = p->next;

//找出的话就打印学生信息

if (num == p->num)

{

printf("num:%d\n", p->num);

printf("name:%s\n", p->name);

printf("sex:%s\n", p->sex);

printf("major:%s\n\n", p->major);

}

else

printf("此学号下无对应学生\n\n");

}

if (choice == 2)

{

printf("请输入你要查找学生的姓名:\n");

scanf\_s("%s", name, 10);

//遍历链表试图找出学生

while (p->next != NULL && strcmp(name, p->name) != 0)

p = p->next;

//找出的话打印学生姓名

if (strcmp(name, p->name) == 0)

{

printf("num:%d\n", p->num);

printf("name:%s\n", p->name);

printf("sex:%s\n", p->sex);

printf("major:%s\n\n", p->major);

}

else

printf("无此学生\n\n");

}

}

//修改学生信息

void Correct(struct stu \*head)

{

struct stu \*p = head;

int num;

if (p == NULL)

exit(0);

printf("请输入你要修改学生的学号:");

scanf\_s("%d", &num);

//遍历链表试图找出这个学生

while (p->next != NULL && num != p->num)

p = p->next;

//如果找出的话

if (num == p->num)

{

printf("请输入你要修改的名字:");

scanf\_s("%s", p->name, 10);

printf("请输入你要修改的性别:");

scanf\_s("%s", p->sex, 10);

printf("请输入你要修改的专业:");

scanf\_s("%s", p->major, 10);

printf("修改成功");

}

else

printf("此学号下无对应学生存在\n\n");

}

//销毁链表

void Free(struct stu \*head)

{

struct stu \*p = head, \*pr = NULL;//pr作为被释放掉的指针

while (p != NULL)

{

pr = p;

p = p->next;

free(pr);

}

}