#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <windows.h>

#include <time.h>

#include <io.h>

#define MAX 50

struct user

{

char user\_Name[6];

char user\_Pass[6];

} stu[1] = { "admin", "admin" }; //初始化用户名和对应密码

void print\_Passwd(void)

{

int i = 0;//当前账号标记

int flag1 = 0;//登录成功标记

int flag = 0;//密码正确标记

char p[6];//临时用户名

char s[6];//临时密码

char num = 0;//密码次数

char temp;//登录后改密码

printf("请输入登录名:\n");

scanf("%s", p);

if (strcmp(stu[i].user\_Name, p) == 0)

{

flag = 1;

}

if (1 == flag)

{

printf("请输入密码:\n");

scanf("%s", s);

}

else

{

printf("没有此用户:\n");

exit(0);

}

while (strcmp(stu[i].user\_Pass, s) != 0)

{

printf("密码错误!\n");

printf("请重新输入密码:\n");

scanf("%s", s);

}

printf("登录成功!\n");

printf("注:你是管理员!\n");

}

typedef struct node1

{

int school; /\*院系编号\*/

int record; /\*项目成绩\*/

struct node1 \*next; /\*链域\*/

} Schools;

typedef struct

{

int item; /\*项目编号\*/

Schools \*firstschool; /\*链域指向链表中第一个结点\*/

} ITEM;

typedef struct

{

int z; /\* 项目总数 \*/

ITEM a[MAX];

} ALLitems;

typedef struct node2

{

int item; /\*该院系获奖的项目\*/

int record; /\*项目成绩\*/

struct node2 \*next; /\*链域\*/

} Items;

typedef struct

{

int school; /\*院系编号\*/

int score; /\*院系总分\*/

int boys; /\*男团体总分\*/

int girls; /\*女团体总分\*/

Items \*firstitem; /\*链域指向链表中第一个获奖项目的结点\*/

} SCHNode;

typedef struct

{

int n; /\* 院系总数 \*/

SCHNode b[MAX];

} ALLNode;

ALLitems \*g1;

ALLNode \*g2;

void funct1(ALLitems \*g1, ALLNode \*g2)

{

Schools \*p1;

Items \*p2;

int i = 1, j, k, c, m, w, h, x;

p1 = (Schools \*)malloc(sizeof(Schools));

p2 = (Items \*)malloc(sizeof(Items));

if (!p1 || !p2)

exit(1);

printf("\n \*\*\*\*\*\*输入各个项目信息 \*\*\*\*\*\* \n\n");

printf(" 输入男子项目总数m:");

scanf("%d", &m);

if (m < 0 || m>20)

{

printf(" 输入有误,m是20以内的整数,请重新输入:");

scanf("%d", &m);

}

printf(" 输入女子项目总数w:");

scanf("%d", &w);

if (w < 0 || w>20)

{

printf(" 输入有误,w是20以内的整数,请重新输入:");

scanf("%d", &w);

}

printf(" 输入参加运动会的院系总数n:");

scanf("%d", &g2->n);

if (g2->n < 0 || g2->n>20)

{

printf(" 输入有误,n是20以内的整数,请重新输入:");

scanf("%d", &g2->n);

}

g1->z = m + w;

printf(" 则项目编号为男子1-%d,女子%d-%d", m, m + 1, g1->z);

printf("\n\n \*\*\*\*记录运动会成绩\*\*\*\*");

printf("\n\n (输入0标志结束)\n");

printf("\t项目1:50米、项目2:100米、项目3:200米、项目4:400米、项目5:1500米、\n");

printf("\t项目6:跳高、项目7:跳远 、项目8:标枪 、项目9:铅球 、项目10:铁饼\n");

for (k = 1; k <= g1->z; k++)

{

g1->a[k].item = k;

g1->a[k].firstschool = NULL;

}

for (k = 1; k <= g2->n; k++)

{

g2->b[k].school = k;

g2->b[k].firstitem = 0;

g2->b[k].score = 0;

g2->b[k].boys = 0;

g2->b[k].girls = 0;

}

g2->b[0].score = 0;

g2->b[0].boys = 0;

g2->b[0].girls = 0;

while (i != 0)

{

printf("\n 项目:");

scanf("%d", &i);

if (i != 0&&i>=1&&i<=m+w)

{

printf(" 请输入该项目的名次取法（1-3）：\n");

printf(" 1.取前三名，分别得分5.3.2\n");

printf(" 2.取前五名，分别得分7.5.3.2.1\n");

printf(" 3.用户自定义，各名次权值由用户指定\n");

scanf("%d",&h);

if(h<=3&&h>=1)

{

switch(h)

{

case 1:

printf(" 请输入前三名：\n");

h=3;

do

{

printf(" 第%d名:院系(院系编号为数字)", h);

scanf("%d", &x);

p1 = (Schools \*)malloc(sizeof(Schools));

p1->school = x;

p2 = (Items \*)malloc(sizeof(Items));

p2->item = i;

if (h == 3) p2->record = p1->record = 2;

if (h == 2) p2->record = p1->record = 3;

if (h == 1) p2->record = p1->record = 5;

p1->next = g1->a[i].firstschool;

g1->a[i].firstschool = p1;

p2->next = g2->b[x].firstitem;

g2->b[x].firstitem = p2;

g2->b[x].score = g2->b[x].score + p2->record; /\* 累计总分 \*/

if (i <= m) g2->b[x].boys = g2->b[x].boys + p2->record; /\* 累计男团体总分 \*/

else g2->b[x].girls = g2->b[x].girls + p2->record; /\* 累计女团体总分 \*/

h--;

}

while (x != 0 && h != 0);

break;

case 2:

printf(" 请输入前五名：\n");

h=5;

do

{

printf(" 第%d名:院系(院系编号为数字)", h);

scanf("%d", &x);

p1 = (Schools \*)malloc(sizeof(Schools));

p1->school = x;

p2 = (Items \*)malloc(sizeof(Items));

p2->item = i;

if (h == 5) p2->record = p1->record = 1;

if (h == 4) p2->record = p1->record = 2;

if (h == 3) p2->record = p1->record = 3;

if (h == 2) p2->record = p1->record = 5;

if (h == 1) p2->record = p1->record = 7;

p1->next = g1->a[i].firstschool;

g1->a[i].firstschool = p1;

p2->next = g2->b[x].firstitem;

g2->b[x].firstitem = p2;

g2->b[x].score = g2->b[x].score + p2->record; /\* 累计总分 \*/

if (i <= m) g2->b[x].boys = g2->b[x].boys + p2->record; /\* 累计男团体总分 \*/

else g2->b[x].girls = g2->b[x].girls + p2->record; /\* 累计女团体总分 \*/

h--;

}

while (x != 0 && h != 0);

break;

case 3:

printf(" 请输入您定义的要取的名次：\n");

scanf("%d",&h);

int a[20];

printf(" 请分别输入第%d-1名所能获得的分值：\n",h);

for(c=h; c>0; c--)

scanf("%d",&a[c]);

do

{

printf(" 第%d名:院系(院系编号为数字)", h);

scanf("%d", &x);

p1 = (Schools \*)malloc(sizeof(Schools));

p1->school = x;

p2 = (Items \*)malloc(sizeof(Items));

p2->item = i;

while(h>0)

{

p2->record=p1->record=a[c];

}

p1->next = g1->a[i].firstschool;

g1->a[i].firstschool = p1;

p2->next = g2->b[x].firstitem;

g2->b[x].firstitem = p2;

g2->b[x].score = g2->b[x].score + p2->record; /\* 累计总分 \*/

if (i <= m) g2->b[x].boys = g2->b[x].boys + p2->record; /\* 累计男团体总分 \*/

else g2->b[x].girls = g2->b[x].girls + p2->record; /\* 累计女团体总分 \*/

h--;

c--;

}

while (x != 0 && h != 0);

break;

default :

break;

}

}

else

printf("输入错误，请重新输入！");

}

else

printf("输入错误，请重新输入！");

}

}

void save()

{

FILE \*fp1, \*fp2;

fp1 = (FILE \*)malloc(sizeof(FILE));

fp2 = (FILE \*)malloc(sizeof(FILE));

if ((fp1 = fopen("sports1.txt", "wb")) == NULL)

{

printf("cannot open file.\n");

return;

}

if (fwrite(g1, sizeof(ALLitems), 1, fp1) != 1)

printf("file write error.\n");

fclose(fp1);

if ((fp2 = fopen("sports2.txt", "wb")) == NULL)

{

printf(" cannot open file.\n");

return;

}

if (fwrite(g2, sizeof(ALLNode), 1, fp2) != 1)

printf("file write error.\n");

fclose(fp2);

}

void funct2(ALLNode \*g2) /\* 输出各院系总分 \*/

{

int k;

printf("\n\n \*\*\*\*\*\* 输出各院系总分 \*\*\*\*\*\*\n");

printf(" 院系编号\t 总分 \n");

for (k = 1; k <= g2->n; k++)

printf(" %d\t\t\t %d\n", k, g2->b[k].score);

printf("\n");

system("pause");

printf(" 按任意键返回主菜单......");

getchar();

}

void funct3(ALLNode \*g2) /\* 按院系编号排序输出 \*/

{

int k;

Items \*p2;

p2 = (Items \*)malloc(sizeof(Items));

printf("\n\n \*\*\*\*\*\* 按院系编号排序输出 \*\*\*\*\*\*\n");

printf(" 院系编号\t\t\t获奖情况 \n");

printf("请按1进行输出：");

scanf("%d",&k);

for (k = 1; k <= g2->n; k++)

{

printf(" %d\t", k);

p2 = g2->b[k].firstitem;

while (p2 != NULL)

{

printf(" 项目%d:得%d分 ", p2->item, p2->record);

p2 = p2->next;

}

printf("\n");

}

printf("\n");

system("pause");

printf("按任意键返回主菜单......");

getchar();

}

void funct4(ALLNode \*g2) /\* 按院系总分排序输出 \*/

{

int i, j, k;

Items \*p2;

printf("\n\n \*\*\*\*\*\* 按院系总分排序输出 \*\*\*\*\*\*\n");

printf(" 院系编号\t\t总分 \n");

printf("请按1进行输出：");

scanf("%d",&k);

for (i = 2; i <= g2->n; i++)

{

p2 = g2->b[k].firstitem;

printf("\n");

g2->b[0].score = g2->b[i].score;

g2->b[0].boys = g2->b[i].boys;

g2->b[0].girls = g2->b[i].girls;

g2->b[0].school = g2->b[i].school;

j = i - 1;

while (g2->b[0].score<g2->b[j].score&&j>0)

{

g2->b[j + 1].score = g2->b[j].score;

g2->b[j + 1].boys = g2->b[j].boys;

g2->b[j + 1].girls = g2->b[j].girls;

g2->b[j + 1].school = g2->b[j].school;

j--;

}

g2->b[j + 1].score = g2->b[0].score;

g2->b[j + 1].boys = g2->b[0].boys;

g2->b[j + 1].girls = g2->b[0].girls;

g2->b[j + 1].school = g2->b[0].school;

}

for (k = 1; k <= g2->n; k++)

printf("%d \t\t%d\n", g2->b[k].school, g2->b[k].score);

system("pause");

printf("按任意键返回主菜单......");

getchar();

}

void funct5(ALLNode \*g2) /\* 按男团体总分排序输出 \*/

{

int i, j, k;

Items \*p2;

p2 = (Items \*)malloc(sizeof(Items));

printf("\n\n \*\*\*\*\*\* 按男团体总分排序输出 \*\*\*\*\*\*\n");

printf("院系编号\t\t男团体总分 \n");

printf("请按1进行输出：");

scanf("%d", &k);

for (i = 2; i <= g2->n; i++)

{

p2 = g2->b[k].firstitem;

printf("\n");

g2->b[0].score = g2->b[i].score;

g2->b[0].boys = g2->b[i].boys;

g2->b[0].girls = g2->b[i].girls;

g2->b[0].school = g2->b[i].school;

j = i - 1;

while (g2->b[0].boys<g2->b[j].boys&&j>0)

{

g2->b[j + 1].score = g2->b[j].score;

g2->b[j + 1].boys = g2->b[j].boys;

g2->b[j + 1].girls = g2->b[j].girls;

g2->b[j + 1].school = g2->b[j].school;

j--;

}

g2->b[j + 1].score = g2->b[0].score;

g2->b[j + 1].boys = g2->b[0].boys;

g2->b[j + 1].girls = g2->b[0].girls;

g2->b[j + 1].school = g2->b[0].school;

}

for (k = 1; k <= g2->n; k++)

printf("%d\t\t %d\n", g2->b[k].school, g2->b[k].boys);

system("pause"); // 暂停等待

printf("按任意键返回主菜单......");

getchar();

}

void funct6(ALLNode \*g2) /\* 按女团体总分排序输出 \*/

{

int i, j, k;

Items \*p2;

p2 = (Items \*)malloc(sizeof(Items));

printf("\n\n \*\*\*\*\*\* 按女团体总分排序输出 \*\*\*\*\*\*\n");

printf("院系编号\t\t女团体总分 \n");

printf("请按1进行输出：");

scanf("%d",&k);

for (i = 2; i <= g2->n; i++)

{

p2 = g2->b[k].firstitem;

printf("\n");

g2->b[0].score = g2->b[i].score;

g2->b[0].boys = g2->b[i].boys;

g2->b[0].girls = g2->b[i].girls;

g2->b[0].school = g2->b[i].school;

j = i - 1;

while (g2->b[0].girls<g2->b[j].girls&&j>0)

{

g2->b[j + 1].score = g2->b[j].score;

g2->b[j + 1].boys = g2->b[j].boys;

g2->b[j + 1].girls = g2->b[j].girls;

g2->b[j + 1].school = g2->b[j].school;

j--;

}

g2->b[j + 1].score = g2->b[0].score;

g2->b[j + 1].boys = g2->b[0].boys;

g2->b[j + 1].girls = g2->b[0].girls;

g2->b[j + 1].school = g2->b[0].school;

}

for (k = 1; k <= g2->n; k++)

printf("%d\t\t %d\n", g2->b[k].school, g2->b[k].girls);

system("pause");

printf("按任意键返回主菜单......");

getchar();

}

void funct7(ALLNode \*g2) /\* 按院系编号查询院系某个项目情况 \*/

{

int i, j;

Items \*p2;

printf("\n \*\*\*\*\*\* 按院系编号查询院系某个项目情况 \*\*\*\*\*\*\n");

printf(" 输入要查询的院系编号:");

scanf("%d", &i);

printf(" 输入要查询的项目编号:");

scanf("%d", &j);

p2 = g2->b[i].firstitem;

while (p2 != NULL)

{

if (p2->item == j)

printf(" 院系编号:%d\t项目%d:得%d分\n", i, p2->item, p2->record);

p2 = p2->next;

}

printf("\n");

system("pause");

printf(" 按任意键返回主菜单......");

getchar();

}

void funct8(ALLitems \*g1) /\* 按项目编号查询取得名次的院系 \*/

{

int i;

Schools \*p1;

printf("\n\*\*\* 按项目编号查询取得名次的院系 \*\*\*\n");

printf("输入要查询的项目编号:");

scanf("%d", &i);

printf("项目编号\t\t\t取得名次的院系\n");

printf("%d\t", i);

p1 = g1->a[i].firstschool;

while (p1 != NULL)

{

printf(" 学校%d:得%d分 ", p1->school, p1->record);

p1 = p1->next;

}

printf("\n\n");

system("pause");

printf("按任意键返回主菜单......");

getchar();

}

void funct9(ALLitems \*g1, ALLNode \*g2)

{Schools \*p1;

Items \*p2;

int p,q;

printf ("请输入要删除的学院编号以及项目：\n");

scanf ("%d %d",&p,&q);

g2->b[p].firstitem=p2;

while (1){

if (p2->item==q)

{

p2->record=0;

break;

}

else

if (p2->next!=NULL)

p2=p2->next;

else break;

}

g1->a[q].firstschool = p1;

while (1){

if (p1->school==p)

{

p1->record=0;

break;

}

else

if (p1->next!=NULL)

p1=p1->next;

else break;

}

system("pause");

printf("按任意键返回主菜单......");

getchar();

}

void main()

{

system("color 3f");

int temp;

printf("请按1进入登陆界面\n");

scanf("%d", &temp);

if (1 == temp)

{

int t;

void print\_Passwd(void);

while (1)

{

print\_Passwd();

break;

}

g2 = (ALLNode\*)malloc(sizeof(ALLNode));

g1 = (ALLitems\*)malloc(sizeof(ALLitems));

if (!g2 || !g1)

exit(1);

system("cls");

while(1)

{

for (;;)

{

struct tm \*pt; /\*定义时间结构体\*/

time\_t t;

t=time(NULL);

pt=localtime(&t); /\*读取系统日期并把它放到结构体中\*/

printf("\t\t\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*\n");

printf("\t\t 当前系统日期:%d-%d-%d\n",pt->tm\_year+1900,pt->tm\_mon+1,pt->tm\_mday); /\*显示当前系统日期\*/

printf("\t\t\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*\n");

printf("\t\t 运动会分数统计系统 \n");

printf("\t\t\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*\n");

printf("\t\t\* 1.输入各个项目成绩并存储文件 \*\t \n");

printf("\t\t\* 2.统计各院系总分 \*\t \n");

printf("\t\t\* 3.按院系编号排序输出 \*\t \n");

printf("\t\t\* 4.按院系总分排序输出 \*\t \n");

printf("\t\t\* 5.按男团体总分排序输出 \*\t \n");

printf("\t\t\* 6.按女团体总分排序输出 \*\t \n");

printf("\t\t\* 7.按院系编号查询院系某个项目情况 \*\t \n");

printf("\t\t\* 8.按项目编号查询取得名次的院系 \*\t \n");

printf("\t\t\* 9.删除成绩 \*\t \n");

printf("\t\t\* 10.修改成绩 \*\t \n");

printf("\t\t\* 0.退出 \*\n");

printf("\t\t\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*\n");

printf("\t\t \n");

printf("\t\t \n");

printf("\t\t \n");

printf("\t\t 请选择(0-10):");

scanf("%d", &t);

system("cls");

switch (t)

{

case 1:

funct1(g1, g2);

save();

break;

case 2:

funct2(g2);

break;

case 3:

funct3(g2);

break;

case 4:

funct4(g2);

break;

case 5:

funct5(g2);

break;

case 6:

funct6(g2);

break;

case 7:

funct7(g2);

break;

case 8:

funct8(g1);

break;

case 9:

funct9(g1,g2);

break;

case 0:

exit(0);

default:

{

printf("输入有误,请重新选择(按任意键返回）:");

getchar();

system("cls");

break;

}

}

}

system("cls");

}

}

}