פרויקט 1

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <string.h>

//סטראקטים

typedef struct headType

{

struct Team\* teamList;

struct PlayerNode\* playerList;

}headder, \* headPtr;

typedef struct Player

{

char playerId[20];

char firstName[15];

char lastName[20];

int age;

}playerRec, \* playerPtr;

typedef struct PlayerNode

{

struct Player PL;

struct Team\* tmptr;

struct PlayerNode\* next;

}playerNodeRec, \* playerNodePtr;

typedef struct Team

{

char teamName[20];

int num;

playerPtr\* players;

struct Team\* next;

}teamRec, \* teamPtr;

//משתנים גלובליים

headder head\_no\_ptr;

headPtr head = &head\_no\_ptr;

//פונקציה המשווה בין שם המשפחה של שחקנים

int compare\_players(Player p1, Player p2)

{

return strcmp(p1.lastName, p2.lastName);

}

//פונקציה המחליפה בין ערכי שני שחקנים

void switch\_compared\_players(playerPtr p1, playerPtr p2)

{

Player temp = { \*p1->playerId,\*p1->firstName,\*p1->lastName,p1->age };

p1->age = p2->age;

strcpy(p1->firstName, p2->firstName);

strcpy(p1->lastName, p2->lastName);

strcpy(p1->playerId, p2->playerId);

p2->age = temp.age;

strcpy(p2->firstName, temp.firstName);

strcpy(p2->lastName, temp.lastName);

strcpy(p2->playerId, temp.playerId);

}

//פעולה המחליפה בין ערכי קבוצות

void switch\_teams(teamPtr t1, teamPtr t2)

{

teamPtr temp = (teamPtr)malloc(sizeof(teamRec));

temp->num = t1->num;

temp->players = t1->players;

strcpy(temp->teamName, t1->teamName);

t1->num = t2->num;

t1->players = t2->players;

strcpy(t1->teamName, t2->teamName);

t2->num = temp->num;

t2->players = temp->players;

strcpy(t2->teamName, temp->teamName);

free(temp);

}

//פעולה המקבלת פוינטר מסוג הד ואיי די מסוג סטרינג ומחזירה את הצומת של השחקן עם האיי די הזה

playerNodePtr find\_player(char id[]) {

playerNodePtr p = head->playerList;

while (p)

{

if (strcmp(p->PL.playerId, id) == 0)

return p;

p = p->next;

}

return NULL;

}

//פעולה המוצאת את הצומת של קבוצה ומחזירה אותה

teamPtr find\_team(char name[])

{

teamPtr tPtr = head->teamList;

while (tPtr)

{

if (strcmp(tPtr->teamName, name) == 0)

return tPtr;

tPtr = tPtr->next;

}

return NULL;

}

//פונקציה אשר מוחקת שחקן מקבוצה

int delete\_player\_from\_team(char id[], char name[])

{

teamPtr t = find\_team(name);

playerNodePtr p = find\_player(id);

if ((p == NULL) || (p->tmptr == NULL)) return 0;

else {

int i, x = t->num;

for (i = 0; i < x; i++)

{

if (t->players[i]->playerId == id)

{

x = i; break;

}

}

switch\_compared\_players(t->players[i], t->players[t->num]);

t->num--;

t->players = (playerPtr\*)realloc(t->players, (t->num) \* sizeof(playerRec));

}

return 1;

}

//פעולה אשר קולטת סטרינג ומחזירה אותו באופן דינאמי

char\* getStr()

{

char\* line = NULL, \* tmp = NULL;

int size = 0, index = 0;

int ch = EOF;

while (ch) {

ch = getc(stdin);

if (ch == EOF || ch == ' ')

{

ch = 0;

}

if (size <= index && ch != '\n')

{

size += 1;

tmp = (char\*)realloc(line, size);

if (!tmp) {

free(line);

line = NULL;

break;

}

line = tmp;

}

if (ch != '\n')

line[index++] = ch;

}

return line;

}

//פונקציה שמוסיפה קבוצה ריקה

void add\_team()

{

teamPtr t = (teamPtr)malloc(sizeof(teamRec));

printf("enter team's details: name\n");

char str[20];

scanf(" %s", str);

strcpy(t->teamName, str);

t->num = 0;

teamPtr tPtr = find\_team(t->teamName);

if (tPtr)

printf("-----ERROR\n");

else {

t->next = head->teamList;

head->teamList = t;

t->players = NULL;

printf("SUCCESS\n");

}

}

//פןנקציה הקולטת פרטי שחקן ופוינטר של הד ומוסיה את השחקן בתחילת רשימת השחקנים

void add\_player()

{

playerPtr p = (playerPtr)malloc(sizeof(playerRec));

printf("enter player's details: id , first name, last name, age\n");

strcpy(p->playerId, getStr());

strcpy(p->firstName, getStr());

strcpy(p->lastName, getStr());

scanf(" %d", &p->age);

playerNodePtr pPtr = find\_player(p->playerId);

if (pPtr)

printf("ERROR\n");

else

{

pPtr = (playerNodePtr)malloc(sizeof(playerNodeRec));

pPtr->tmptr = NULL;

pPtr->PL = \*p;

pPtr->next = head->playerList;

head->playerList = pPtr;

printf("SUCCESS\n");

}

}

//פונקציה המקבלת פוינטר של הד ואיי די מסוג סטרינג ומוחקת את השחקן עם האיי די הזה

int delete\_player(char id[])

{

playerNodePtr p, q;

int status = 1;

p = find\_player(id);

if (p == NULL) status = 0;

else

{

if (p->tmptr)

status = delete\_player\_from\_team(id, p->tmptr->teamName);

q = head->playerList;

if (head->playerList == p)

{

head->playerList = q->next;

free(p);

}

else

{

while (q->next)

{

if (q->next == p) {

q->next = p->next;

free(p);

break;

}

q = q->next;

}

}

}

return status;

}

//פןנקציה אשר מוחקת קבוצה והופכץ אץ דחקניה לחופשיים

int delete\_team(char name[])

{

teamPtr t, s;

int status = 1;

int i;

t = find\_team(name);

if (t == NULL) status = 0;

else

{

/\*for (i = 0; i < t->num; i++);

{

playerNodePtr pPtr = find\_player(t->players[i]->playerId);

pPtr->tmptr = NULL;

}\*/

s = head->teamList;

if (s == t)

{

head->teamList = head->teamList->next;

}

else

{

while (s->next != t)

s = s->next;

s->next = t->next;

}

free(t);

}

return status;

}

//פעולה המוסיפה שחקן לקבוצה

int join\_player\_to\_team(char name[], char id[])

{

int status = 1;

playerNodePtr p;

teamPtr t;

p = find\_player(id);

t = find\_team(name);

if (p == NULL) return 0;

else if (t == NULL) return 0;

else if (p->tmptr != NULL) return 0;

else

{

p->tmptr = t;

if (t->num == 0)

{

\*(t->players) = (playerPtr)realloc(t->players, sizeof(playerPtr));

t->players[0] = &p->PL;

t->num = 1;

}

else

{

\*(t->players) = (playerPtr)realloc(t->players, t->num \* sizeof(playerPtr));

t->players[t->num] = &p->PL;

t->num++;

}

}

return status;

}

void main()

{

int decision = 0, flag = 1;

playerNodePtr player\_foundation;

teamPtr team\_foundation;

while (flag)

{

printf("%s", "\npress 1 to add player\npress 2 to locate player\npress 3 to delete player\npress 4 to add team\npress 5 to locate team\npress 6 to delete team\npress 7 to add player to team\npress 8 to delete player from team\npress 9 to print the players in the leage\npress 10 to print all teams\npress 11 to print team's details\npress 12 to sort players in team\npress 13 to sort teams by name\npress 14 to sort teams by amount of players\npress 15 to find out if the player is free\npress 16 to finish the program\n");

scanf("%d", &decision);

switch (decision)

{

case 1: {

add\_player();

break;

}

case 2: {

char id[15];

printf("enter player's id: ");

scanf(" %s", &id);

player\_foundation = find\_player(id);

break;

}

case 3: {

char id[15];

printf("enter player's id: ");

scanf("%s", &id);

printf("\n");

int x = delete\_player(id);

if (x == 1)

printf("Success\n");

else

printf("Error\n");

break;

}

case 4: {

add\_team();

break;

}

case 5: {

char name[15];

printf("enter team's name: ");

scanf(" %s", &name);

printf("\n");

team\_foundation = find\_team(name);

break;

}

case 6: {

char name[15];

printf("enter team's name: ");

scanf("%s", &name);

printf("\n");

int x = delete\_team(name);

if (x == 1)

printf("Success\n");

else

printf("Error\n");

break;

}

case 7: {

char name[15];

printf("enter team's name: ");

scanf("%s", &name);

printf("\n");

char id[15];

printf("enter players's id: ");

scanf("%s", &id);

printf("\n");

int x = join\_player\_to\_team(name, id);

if (x == 1)

printf("Success\n");

else

printf("Error\n");

break;

}

case 8: {

char name[15];

printf("enter team's name: ");

scanf("%s\n", &name);

char id[15];

printf("enter players's id: ");

scanf("%s\n", &id);

int x = delete\_player\_from\_team(id, name);

if (x == 1)

printf("Success\n");

else

printf("Error\n");

break;

}

case 9: {

playerNodePtr p = head->playerList;

if (!p)

printf("No Players\n");

while (p)

{

printf("id: %s name: %s %s age: %d\n", p->PL.playerId, p->PL.firstName, p->PL.lastName, p->PL.age);

p = p->next;

}

break;

}

case 10: {

teamPtr t = head->teamList;

if (!t)

printf("No teams\n");

else printf("team's names:");

while (t)

{

printf(" %s \t", t->teamName);

t = t->next;

}

printf("\n");

break;

}

case 11: {

char name[15];

printf("enter team's name: ");

scanf("%s\n", &name);

teamPtr t = find\_team(name);

if (t == NULL) { printf("Error"); break; }

else {

for (int i = 0; i < t->num; i++)

printf("id: %s name: %s %s age: %d\n", t->players[0]->playerId, t->players[0]->firstName, t->players[0]->lastName, t->players[0]->age);

}

break;

}

case 12: {

char name[15];

printf("enter team's name: ");

scanf("%s\n", &name);

teamPtr t = find\_team(name);

if (t == NULL) { printf("Error"); break; }

else

{//ממיין

int size = sizeof(t->players) / sizeof(Player);

for (int a = 0; a < size; a++)

for (int b = 1; b < size; b++)

{

if (compare\_players(\*t->players[a], \*t->players[b]) > 0)

switch\_compared\_players(t->players[a], t->players[b]);

}

//מדפיס

teamPtr t = find\_team(name);

if (t == NULL) { printf("Error"); break; }

else {

for (int i = 0; i < t->num; i++)

printf("id: %s name: %s %s age: %d\n", t->players[0]->playerId, t->players[0]->firstName, t->players[0]->lastName, t->players[0]->age);

}

break;

}

}

case 13: {

teamPtr t1 = head->teamList;

if (t1 == NULL) { printf("ERROR"); break; }

else

{

teamPtr t2 = head->teamList->next;

if (t2 == NULL)break;

else

{

while (t1)

{

t2 = t1->next;

while (t2)

{

if (strcmp(t1->teamName, t2->teamName) > 0)

switch\_teams(t1, t2);

t2 = t2->next;

}

t1 = t1->next;

}

}

}

}

case 14: {

teamPtr t1 = head->teamList;

if (t1 == NULL) { printf("ERROR"); break; }

else

{

teamPtr t2 = head->teamList->next;

if (t2 == NULL)break;

else

{

while (t1)

{

t2 = t1->next;

while (t2)

{

if (t1->num > t2->num)

switch\_teams(t1, t2);

t2 = t2->next;

}

t1 = t1->next;

}

}

}

}

case 15: {

char id[15]; int flag = 1;

printf("enter players's id: ");

scanf("%s", &id);

printf("\n");

playerNodePtr p = find\_player(id);

if (p == NULL)

{

printf("Error"); break;

}

if (p->tmptr == NULL)

{

printf("free player"); break;

}

teamPtr t = p->tmptr;

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

printf("id: %s name: %s %s age: %d\n", t->players[0]->playerId, t->players[0]->firstName, t->players[0]->lastName, t->players[0]->age);

}

break;

}

case 16: {

playerNodePtr p = head->playerList;

head->playerList = NULL;

teamPtr t = head->teamList;

head->teamList = NULL;

while (p)

delete\_player(p->PL.playerId);

while (t)

delete\_team(t->teamName);

printf("Bye");

flag = 0;

break;

}

}

}

}