#include<stdlib.h>

#include<string.h>

#include<stdio.h>

int main(){

int select;

printf("\t\t\t WELCOME TO KABBADI STATS -ENTER YOUR CHOICE \n");

printf("\t\t1.Raider\n\t\t2.Defender\n\t\t3.All Rounder\n");

scanf("%d",&select);

while(select!=4){

switch(select){

case 1:

main1();

break;

case 2:

main2();

break;

case 3:

main3();

break;

default:

printf("invalid");

}

}

}

struct Raider

{

int jersey;

char player\_name[100];

float success;

struct Raider \*next;

};

struct Raider \* insert(struct Raider \*head, int jersey, char player\_name[100], float success)

{

struct Raider \* raider = (struct Raider \*) malloc(sizeof(struct Raider));

raider->jersey = jersey;

strcpy(raider->player\_name, player\_name);

raider->success = success;

raider->next = NULL;

if(head==NULL)

{

head = raider;

}

else{

raider->next = head;

head = raider;

}

return head;

}

void search(struct Raider \*head, int jersey)

{

struct Raider \* temp = head;

while(temp!=NULL){

if(temp->jersey==jersey){

printf("Player\_name: %s\n", temp->player\_name);

printf("Jersey: %d\n", temp->jersey);

printf("Success: %0.4f\n", temp->success);

return;

}

temp = temp->next;

}

printf("Raider with jersey %d is not found !!!\n", jersey);

}

void update(struct Raider \*head, int jersey)

{

struct Raider \* temp = head;

while(temp!=NULL){

if(temp->jersey==jersey){

printf("Record with jersey %d Found !!!\n", jersey);

printf("Enter new player\_name: ");

scanf("%s", temp->player\_name);

printf("Enter new success: ");

scanf("%f",&temp->success);

printf("Updation Successful!!!\n");

return;

}

temp = temp->next;

}

printf("Raider with jersey %d is not found !!!\n", jersey);

}

struct Raider \* Delete(struct Raider \*head, int jersey)

{

struct Raider \* temp1 = head;

struct Raider \* temp2 = head;

while(temp1!=NULL){

if(temp1->jersey==jersey){

if(temp1==temp2){

head = head->next;

free(temp1);

}

else{

temp2->next = temp1->next;

free(temp1);

}

return head;

}

temp2 = temp1;

temp1 = temp1->next;

}

return head;

}

void display(struct Raider \*head)

{

struct Raider \* temp = head;

while(temp!=NULL){

printf("Jersey: %d\n", temp->jersey);

printf("Player\_name: %s\n", temp->player\_name);

printf("Success: %0.4f\n\n", temp->success);

temp = temp->next;

}

}

struct Raider \* sort(struct Raider \*head){

struct Raider \*temp = NULL;

struct Raider \*ptr;

struct Raider \*lptr;

while(head!=NULL){

ptr = head;

lptr = head;

while(ptr!=NULL){

if(ptr->jersey > lptr->jersey){

lptr = ptr;

}

ptr = ptr->next;

}

temp = insert(temp, lptr->jersey, lptr->player\_name, lptr->success);

head = Delete(head, lptr->jersey);

}

return temp;

}

int main1()

{

struct Raider \*head = NULL;

int choice;

char player\_name[100];

int jersey;

float success;

printf("1 To insert raider details\n2 To search for raider details\n3 To remove raider due to weak performance\n4 To substitute raider due to injury\n5 To view all raider details\n7.TO SKIP TO MAIN MENU");

do

{

printf("\nEnter Choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter player\_name: ");

scanf("%s", player\_name);

printf("Enter jersey number: ");

scanf("%d", &jersey);

printf("Enter success rate of raider: ");

scanf("%f", &success);

head = insert(head,jersey, player\_name , success);

break;

case 2:

printf("Enter jersey number to search: ");

scanf("%d", &jersey);

search(head,jersey);

break;

case 3:

printf("Enter jersey number to delete: ");

scanf("%d", &jersey);

head = Delete(head, jersey);

break;

case 4:

printf("Enter jersey number to substitute: ");

scanf("%d", &jersey);

update(head, jersey);

break;

case 5:

display(head);

break;

case 6:

head = sort(head);

break;

case 7:

main();

break;

}

} while (choice != 0);

}

struct Defender

{

int jersey;

char player\_name[100];

float success;

struct Defender \*next;

};

struct Defender \* defend(struct Defender \*head, int jersey, char player\_name[100], float success)

{

struct Defender \* defender = (struct Defender \*) malloc(sizeof(struct Defender));

defender->jersey = jersey;

strcpy(defender->player\_name, player\_name);

defender->success = success;

defender->next = NULL;

if(head==NULL)

{

head = defender;

}

else{

defender->next = head;

head = defender;

}

return head;

}

void find(struct Defender \*head, int jersey)

{

struct Defender \* temp = head;

while(temp!=NULL){

if(temp->jersey==jersey){

printf("player\_name: %s\n", temp->player\_name);

printf("Jersey: %d\n", temp->jersey);

printf("Success: %0.4f\n", temp->success);

return;

}

temp = temp->next;

}

printf("Defender with jersey %d is not found !!!\n", jersey);

}

void modify(struct Defender \*head, int jersey)

{

struct Defender \* temp = head;

while(temp!=NULL){

if(temp->jersey==jersey){

printf("Record with jersey %d Found !!!\n", jersey);

printf("Enter new player\_name: ");

scanf("%s", temp->player\_name);

printf("Enter new success: ");

scanf("%f",&temp->success);

printf("Updation Successful!!!\n");

return;

}

temp = temp->next;

}

printf("Defender with jersey %d is not found !!!\n", jersey);

}

struct Defender \* Erase(struct Defender \*head, int jersey)

{

struct Defender \* temp1 = head;

struct Defender \* temp2 = head;

while(temp1!=NULL){

if(temp1->jersey==jersey){

if(temp1==temp2){

/\* this condition will run if

the record that we need to erase is the first node

of the linked list \*/

head = head->next;

free(temp1);

}

else{

/\* temp1 is the node we need to erase

temp2 is the node previous to temp1 \*/

temp2->next = temp1->next;

free(temp1);

}

return head;

}

temp2 = temp1;

temp1 = temp1->next;

}

return head;

}

void view(struct Defender \*head)

{

struct Defender \* temp = head;

while(temp!=NULL){

printf("Jersey: %d\n", temp->jersey);

printf("Player\_name: %s\n", temp->player\_name);

printf("Success: %0.4f\n\n", temp->success);

temp = temp->next;

}

}

struct Defender \* backup(struct Defender \*head){

struct Defender \*temp = NULL;

struct Defender \*ptr;

struct Defender \*lptr;

while(head!=NULL){

ptr = head;

lptr = head;

while(ptr!=NULL){

if(ptr->jersey > lptr->jersey){

lptr = ptr;

}

ptr = ptr->next;

}

temp = defend(temp, lptr->jersey, lptr->player\_name, lptr->success);

head = Erase(head, lptr->jersey);

}

return temp;

}

int main2()

{

struct Defender \*head = NULL;

int choice;

char player\_name[100];

int jersey;

float success;

printf("1 To insert defender details\n2 To search for defender details\n3 To delete defender details\n4 To modify defender details\n5 To view all defender details\n7.TO SKIP TO MAIN MENU");

do

{

printf("\nEnter Choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter defender\_name: ");

scanf("%s", player\_name);

printf("Enter jersey number: ");

scanf("%d", &jersey);

printf("Enter success rate in percentage: ");

scanf("%f", &success);

head = defend(head,jersey, player\_name , success);

break;

case 2:

printf("Enter jersey number to find: ");

scanf("%d", &jersey);

find(head,jersey);

break;

case 3:

printf("Enter jersey number to remove: ");

scanf("%d", &jersey);

head = Erase(head, jersey);

break;

case 4:

printf("Enter jersey to substitute: ");

scanf("%d", &jersey);

modify(head, jersey);

break;

case 5:

view(head);

break;

case 6:

head = backup(head);

break;

case 7:

main();

break;

}

} while (choice != 0);

}

struct Allrounder

{

int jersey;

char player\_name[100];

float success;

struct Allrounder \*next;

};

struct Allrounder \* add(struct Allrounder \*head, int jersey, char player\_name[100], float success)

{

struct Allrounder \* allrounder = (struct Allrounder \*) malloc(sizeof(struct Allrounder));

allrounder->jersey = jersey;

strcpy(allrounder->player\_name, player\_name);

allrounder->success = success;

allrounder->next = NULL;

if(head==NULL)

{

head = allrounder;

}

else{

allrounder->next = head;

head = allrounder;

}

return head;

}

void look(struct Allrounder \*head, int jersey)

{

struct Allrounder \* temp = head;

while(temp!=NULL){

if(temp->jersey==jersey){

printf("Player\_name: %s\n", temp->player\_name);

printf("Jersey: %d\n", temp->jersey);

printf("Success: %0.4f\n", temp->success);

return;

}

temp = temp->next;

}

printf("Allrounder with jersey %d is not found !!!\n", jersey);

}

void new(struct Allrounder \*head, int jersey)

{

struct Allrounder \* temp = head;

while(temp!=NULL){

if(temp->jersey==jersey){

printf("Record with jersey %d Found !!!\n", jersey);

printf("Enter new player\_name: ");

scanf("%s", temp->player\_name);

printf("Enter new success: ");

scanf("%f",&temp->success);

printf("Updation Successful!!!\n");

return;

}

temp = temp->next;

}

printf("Allrounder with jersey %d is not found !!!\n", jersey);

}

struct Allrounder \* Clean(struct Allrounder \*head, int jersey)

{

struct Allrounder \* temp1 = head;

struct Allrounder \* temp2 = head;

while(temp1!=NULL){

if(temp1->jersey==jersey){

if(temp1==temp2){

head = head->next;

free(temp1);

}

else{

temp2->next = temp1->next;

free(temp1);

}

return head;

}

temp2 = temp1;

temp1 = temp1->next;

}

return head;

}

void seen(struct Allrounder \*head)

{

struct Allrounder \* temp = head;

while(temp!=NULL){

printf("Jersey: %d\n", temp->jersey);

printf("Player\_name: %s\n", temp->player\_name);

printf("Success: %0.4f\n\n", temp->success);

temp = temp->next;

}

}

struct Allrounder \* rearrange(struct Allrounder \*head){

struct Allrounder \*temp = NULL;

struct Allrounder \*ptr;

struct Allrounder \*lptr;

while(head!=NULL){

ptr = head;

lptr = head;

while(ptr!=NULL){

if(ptr->jersey > lptr->jersey){

lptr = ptr;

}

ptr = ptr->next;

}

temp = add(temp, lptr->jersey, lptr->player\_name, lptr->success);

head = Clean(head, lptr->jersey);

}

return temp;

}

int main3()

{

struct Allrounder \*head = NULL;

int choice;

char player\_name[100];

int jersey;

float success;

printf("1 To insert allrounder details\n2 To search for allrounder details\n3 To delete allrounder details\n4 To update allrounder details\n5 To display all allrounder details\n7.TO SKIP TO MAIN MENU");

do

{

printf("\nEnter Choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter allrounder\_name: ");

scanf("%s", player\_name);

printf("Enter jersey number: ");

scanf("%d", &jersey);

printf("Enter success rate in percentage: ");

scanf("%f", &success);

head = add(head,jersey, player\_name , success);

break;

case 2:

printf("Enter jersey number to view details: ");

scanf("%d", &jersey);

look(head,jersey);

break;

case 3:

printf("Enter jersey number to remove: ");

scanf("%d", &jersey);

head = Clean(head, jersey);

break;

case 4:

printf("Enter jersey number to subsitute: ");

scanf("%d", &jersey);

new(head, jersey);

break;

case 5:

seen(head);

break;

case 6:

head = rearrange(head);

break;

case 7:

main();

break;

}

} while (choice != 0);

}