//g00340315

//Ray Mannion Rugby Performance Metric Ltd.

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

//struct pass

//{

//char userName[30];

//char password[30];

// struct pass\* NEXT;

//};// password struct

struct node

{

int IRFUnum;

char firstName[30];

char surname[30];

int age;

float height;

float weight;

char club[30];

char email[30];

char playerPosition[15];

int tacklesMissed;

float metresMade;

struct node\* NEXT;

};

void addPlayer\_atStart(struct node\*\* headptr);

void addPlayer(struct node\* headptr, int searchID);

void display(struct node\* headptr);

void searchPlayerNo(struct node\* headptr, int searchID);

void searchPlayerName(struct node\* headptr, char searchName);

void updateByNum(struct node\* headptr, int searchID);

void deletePlayer\_atStart(struct node\*\* headptr);

void deletePlayer(struct node\* headptr);

int length(struct node\* headptr);

int search(struct node\* headptr, int searchID);

void displayNames(struct node\* headptr);

void printToFile(struct node\* headptr);

//char searchByName(struct node\* headptr, char searchName);

void updateByName(struct node\* headptr, char searchName);

void menu();

void tackleStats(struct node\* headptr);

void metresStats(struct node\* headptr);

void printTackle(struct node\* headptr);

void printMetres(struct node\* headptr);

void main()

{

int i, counter = 0, flag = 0;

char uid[25], pwd[25], s\_uid[][25] = { "ray","tom","mary" };

char s\_pwd[][25] = { "123456","654321","246810" }, ch = 'a';//dummy character in ch

//clrscr();

printf("\n Enter the user id : ");

scanf("%s", uid);

printf("\n Enter the password : ");

i = 0;

while (1)

{

ch = getch();

if (ch == 13)

break;

else if (ch == 8)

{

if (i != 0) //this is for avoiding the ENTER instructions getting deleted

{

printf("\b"); //printing backspace to move cursor 1 pos back\*/

printf("%c", 32);//making the char invisible which is already on console

printf("\b"); //printing backspace to move cursor 1 pos back

i--;

pwd[i] = '\0';

}

else

continue;

}

else

{

putchar('\*');// char - '\*' will be printed instead of the password

pwd[i] = ch;

i++;

}

}

pwd[i] = '\0';

for (i = 0; i <= 2; i++)

{

if ((stricmp(uid, s\_uid[i])) == 0 && (strcmp(pwd, s\_pwd[i])) == 0)

{

flag = 1;

break;

}

}

if (flag)

{

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

menu();// calls menu when username and password are correct

}

else

{

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

}

getch();

}

void addPlayer\_atStart(struct node\*\* headptr)

{

struct node\* newNode;

newNode = (struct node\*)malloc(sizeof(struct node));

printf("Enter Players IRFU No. : ");

scanf("%d", &newNode->IRFUnum);

printf("Enter Players first name : ");

scanf("%s", newNode->firstName);

printf("Enter Players surname : ");

scanf("%s", newNode->surname);

printf("Enter Players Age : ");

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

printf("Enter Players height : ");

scanf("%f", &newNode->height);

printf("Enter Players weight : ");

scanf("%f", &newNode->weight);

printf("Enter Players club : ");

scanf("%s", newNode->club);

printf("Enter Players Email : ");

scanf("%s", newNode->email);

if (strstr(newNode->email,".com")==NULL && strstr(newNode->email, ".") == NULL && strstr(newNode->email, "@") == NULL)

{

printf("INVALID EMAIL\n");

printf("Enter Players Email : ");

scanf("%s", newNode->email);

}

else

{

newNode->email;

}

printf("Enter Players Position : ");

scanf("%s", newNode->playerPosition);

printf("Enter tackles missed per match : ");

scanf("%d", &newNode->tacklesMissed);

printf("Enter metres covered per match : ");

scanf("%f", &newNode->metresMade);

printf("=========================================\n");

newNode->NEXT = \*headptr;

\*headptr = newNode;

}// addPlayer\_atStart

void addPlayer(struct node\* headptr, int searchID)

{

int data;

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

temp = headptr;

while (temp->NEXT != NULL) // go to the last node

{

temp = temp->NEXT;

}

struct node \*newNode;

newNode = (struct node\*)malloc(sizeof(struct node));

newNode->IRFUnum = searchID ;

printf("Enter Players first name : ");

scanf("%s", newNode->firstName);

printf("Enter Players surname : ");

scanf("%s", newNode->surname);

printf("Enter Players Age : ");

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

printf("Enter Players height : ");

scanf("%f", &newNode->height);

printf("Enter Players weight : ");

scanf("%f", &newNode->weight);

printf("Enter Players club : ");

scanf("%s", newNode->club);

printf("Enter Players Email : ");

scanf("%s", newNode->email);

if (strstr(newNode->email, ".com") == NULL && strstr(newNode->email, ".") == NULL && strstr(newNode->email, "@") == NULL)

{

printf("INVALID EMAIL\n");

printf("Enter Players Email : ");

scanf("%s", newNode->email);

}

else

{

newNode->email;

}

printf("Enter Players Position : ");

scanf("%s", newNode->playerPosition);

printf("Enter tackles missed per match : ");

scanf("%d", &newNode->tacklesMissed);

printf("Enter metres covered per match : ");

scanf("%f", &newNode->metresMade);

printf("=========================================\n");

newNode->NEXT = NULL;

temp->NEXT = newNode;

}// addPlayer

void display(struct node\* headptr)

{

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

temp = headptr;

while (temp != NULL)

{

printf(" Player Details \n");

printf("=================================\n");

printf(" IRFU NO : %d\n", temp->IRFUnum);

printf(" Surname : %s\n", temp->surname);

printf(" First name : %s\n", temp->firstName);

printf(" Age: %d\n", temp->age);

printf(" Height : %.2f \n", temp->height);

printf(" Weight : %.2f \n", temp->weight);

printf(" Club : %s\n", temp->club);

printf(" Contact email : %s\n", temp->email);

printf(" Position : %s\n", temp->playerPosition);

printf(" Tackles missed per match : %d\n", temp->tacklesMissed);

printf(" Metres made per match : %.2f \n", temp->metresMade);

printf("==============================\n");

temp = temp->NEXT;

}

}// display

void displayNames(struct node\* headptr)

{

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

printf(" Player List \n");

printf("=================================\n");

temp = headptr;

while (temp != NULL)

{

printf(" Name : %s %s\n", temp->firstName, temp->surname);

temp = temp->NEXT;

}

}// display

void searchPlayerNo(struct node \* headptr, int searchID)

{

struct node \*temp;

temp = headptr;

while (temp != NULL)

{

if (searchID == temp->IRFUnum )

{

printf(" Player Found \n");

printf("=================================\n");

printf(" IRFU NO : %d\n", temp->IRFUnum);

printf(" Surname : %s\n", temp->surname);

printf(" First name : %s\n", temp->firstName);

printf(" Age: %d\n", temp->age);

printf(" Height : %.2f \n", temp->height);

printf(" Weight : %.2f \n", temp->weight);

printf(" Club : %s\n", temp->club);

printf("Contact email : %s\n", temp->email);

printf(" Position : %s\n", temp->playerPosition);

printf(" Tackles missed per match : %d\n", temp->tacklesMissed);

printf(" Metres made per match : %.2f \n", temp->metresMade);

printf("==============================\n");

}

temp = temp->NEXT;

}// while

}// searchPlayerNo

void searchPlayerName(struct node \* headptr, char searchName[])

{

struct node \*temp;

//int i = 0;

//int found = 0;

temp = headptr;

while (temp != NULL)

{

//i++;

if (strcmp(searchName, temp->surname) == 0)

{

//found = i;

printf(" Player Found \n");

printf("=================================\n");

printf(" IRFU NO : %d\n", temp->IRFUnum);

printf(" Surname : %s\n", temp->surname);

printf(" First name : %s\n", temp->firstName);

printf(" Age: %d\n", temp->age);

printf(" Height : %.2f \n", temp->height);

printf(" Weight : %.2f \n", temp->weight);

printf(" Club : %s\n", temp->club);

printf("Contact email : %s\n", temp->email);

printf(" Position : %s\n", temp->playerPosition);

printf(" Tackles missed per match : %d\n", temp->tacklesMissed);

printf(" Metres made per match : %.2f \n", temp->metresMade);

printf("==============================\n");

}

temp = temp->NEXT;

}// while

}// searchPlayerName

void updateByNum(struct node\* headptr, int searchID)

{

struct node \*temp;

//int i = 0;

//int found = 0;

temp = headptr;

if (searchID = temp->IRFUnum)

{

printf(" Player Found \n");

printf("=================================\n");

printf(" IRFU NO : %d\n", temp->IRFUnum);

printf(" Surname : %s\n", temp->surname);

printf(" First name : %s\n", temp->firstName);

printf("Enter Players Age : ");

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

printf("Enter Players height : ");

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

printf("Enter Players weight : ");

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

printf("Enter Players club : ");

scanf("%s", temp->club);

printf("Enter Players Email : ");

scanf("%s", temp->email);

if (strstr(temp->email, ".com") == NULL && strstr(temp->email, ".") == NULL && strstr(temp->email, "@") == NULL)

{

printf("INVALID EMAIL\n");

printf("Enter Players Email : ");

scanf("%s", temp->email);

}

else

{

temp->email;

}

printf("Enter Players Position : ");

scanf("%s", temp->playerPosition);

printf("Enter tackles missed per match : ");

scanf("%d", &temp->tacklesMissed);

printf("Enter metres covered per match : ");

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

printf("==============================\n");

}

temp = temp->NEXT;

}// updateByNum

/\*void updateByName(struct node\* headptr, char searchName)

{

struct node \*temp;

//int i = 0;

//int found = 0;

temp = headptr;

if (strcmp(searchName, temp->surname) == 0)

{

printf(" Player Found \n");

printf("=================================\n");

printf(" IRFU NO : %d\n", temp->IRFUnum);

printf(" Surname : %s\n", temp->surname);

printf(" First name : %s\n", temp->firstName);

printf("Enter Players Age : ");

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

//printf("Enter the Student ID :");

//scanf("%s", newNode->ID);

//strcpy(newNode->ID, ID);

printf("Enter Players height : ");

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

printf("Enter Players weight : ");

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

printf("Enter Players club : ");

scanf("%s", temp->club);

printf("Enter Players Email : ");

scanf("%s", temp->email);

printf("Enter Players Position : ");

scanf("%s", temp->playerPosition);

printf("Enter tackles missed per match : ");

scanf("%d", &temp->tacklesMissed);

printf("Enter metres covered per match : ");

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

printf("==============================\n");

}

temp = temp->NEXT;

}// updateByName\*/

void deletePlayer\_atStart(struct node\*\* headptr, int searchID)

{

struct node \*temp;

temp = \*headptr;

\*headptr = temp->NEXT;

free(temp);

}// deletenode\_atStart

void deletePlayer(struct node\* headptr,int searchID)

{

struct node\* temp;

struct node\* prev\_temp;

temp = headptr;

prev\_temp = headptr;

while (temp->NEXT != NULL)

{

prev\_temp = temp;

temp = temp->NEXT;

}

prev\_temp->NEXT = temp->NEXT;

free(temp);

}// deletenode

int length(struct node\* headptr)

{

int len = 0;

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

temp = headptr;

while (temp != NULL)

{

len++;

temp = temp->NEXT;

}

return len;

}// length

int search(struct node\* headptr, int searchID)

{

struct node \*temp;

int i = 0;

int found = 0;

temp = headptr;

while (temp != NULL) // searches throught nodes looking for id

{

i++;

if (searchID == temp->IRFUnum)

found = i;

temp = temp->NEXT;

}

return found;

}// search

/\*char searchByName(struct node\* headptr, char searchName)

{

struct node \*temp;

int i = 0;

int found = 0;

temp = headptr;

while (temp != NULL)

{

i++;

if ((searchName, temp->surname) == 0)

found = i;

temp = temp->NEXT;

}

return found;

}// searchByName\*/

void tackleStats(struct node\* headptr)

{

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

float zeroTacks=0;

float uThreeTacks = 0;

float uFiveTacks = 0;

float oFiveTacks = 0;

int players=0;

float zeroPer=0;

float uTreePer=0;

float uFvePer=0;

float oFvePer=0;

temp = headptr;

while (temp != NULL)

{

if (temp->tacklesMissed == 0) // gives various count of tackles missed

{

zeroTacks++;

}

else if (temp->tacklesMissed > 0 && temp->tacklesMissed < 3)

{

uThreeTacks++;

}

else if (temp->tacklesMissed >= 3 && temp->tacklesMissed < 5)

{

uFiveTacks++;

}

else if (temp->tacklesMissed > 5)

{

oFiveTacks++;

}

temp = temp->NEXT;

}

players = zeroTacks + uThreeTacks + uFiveTacks + oFiveTacks; // adds up players

zeroPer = (zeroTacks/players) \* 100;

uTreePer = (uThreeTacks / players) \* 100; // get percents of tackles missed

uFvePer = (uFiveTacks / players) \* 100;

oFvePer = (oFiveTacks / players) \* 100;

printf("%.2f percent of player miss zero tackles \n", zeroPer);

printf("%.2f percent of player miss 0 - 3 tackles\n", uTreePer);

printf("%.2f percent of player miss 3 - 5 tackles\n", uFvePer);

printf("%.2f percent of player miss over 5 tackles\n", oFvePer);

}// tackleStats

void metresStats(struct node\* headptr)

{

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

float zeroMetres = 0;

float uTenMetres = 0;

float uTwentyMetres = 0;

float oTwentyMetres = 0;

int players=0;

float zeroPer=0;

float uTenPer=0;

float uTwPer=0;

float oTwPer=0;

temp = headptr;

while (temp != NULL)

{

if (temp->metresMade == 0) // gives various count of metres covered

{

zeroMetres++;

}

else if (temp->metresMade > 0 && temp->metresMade < 10)

{

uTenMetres++;

}

else if (temp->metresMade >= 10 && temp->metresMade < 20)

{

uTwentyMetres++;

}

else if (temp->metresMade > 20)

{

oTwentyMetres++;

}

temp = temp->NEXT;

}

players = zeroMetres + uTenMetres + uTwentyMetres + oTwentyMetres;// add up all players

zeroPer = (zeroMetres/ players) \* 100;

uTenPer = (uTenMetres / players) \* 100; // find percents of metres covered

uTwPer = (uTwentyMetres / players) \* 100;

oTwPer == (oTwentyMetres / players) \* 100;

printf("%.2f percent of player make zero metres per match \n", zeroPer);

printf("%.2f percent of player make 0 - 10 metres per match \n", uTenPer);

printf("%.2f percent of player make 10 - 20 metres per match\n", uTwPer);

printf("%.2f percent of player make over 20 metres per match\n", oTwPer);

}// tackleStats

void printTackle(struct node\* headptr)

{

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

float zeroTacks = 0;

float uThreeTacks = 0;

float uFiveTacks = 0;

float oFiveTacks = 0;

int players = 0;

float zeroPer = 0;

float uTreePer = 0;

float uFvePer = 0;

float oFvePer = 0;

temp = headptr;

FILE\* filep;// creates file to write too

filep = fopen("tackleStats.txt", "w");// allows user to write to the text file

while (temp != NULL)

{

if (temp->tacklesMissed == 0) // gives various count of tackles missed

{

zeroTacks++;

}

else if (temp->tacklesMissed > 0 && temp->tacklesMissed < 3)

{

uThreeTacks++;

}

else if (temp->tacklesMissed >= 3 && temp->tacklesMissed < 5)

{

uFiveTacks++;

}

else if (temp->tacklesMissed > 5)

{

oFiveTacks++;

}

temp = temp->NEXT;

}

players = zeroTacks + uThreeTacks + uFiveTacks + oFiveTacks; // adds up players

zeroPer = (zeroTacks / players) \* 100;

uTreePer = (uThreeTacks / players) \* 100; // get percents of tackles missed

uFvePer = (uFiveTacks / players) \* 100;

oFvePer = (oFiveTacks / players) \* 100;

if (filep != NULL)

{

// prints data to text file

fprintf(filep, " Taclkle Stats \n");

fprintf(filep, "===================================================\n");

fprintf(filep,"%.2f percent of player miss zero tackles \n", zeroPer);

fprintf(filep,"%.2f percent of player miss 0 - 3 tackles\n", uTreePer);

fprintf(filep,"%.2f percent of player miss 3 - 5 tackles\n", uFvePer);

fprintf(filep,"%.2f percent of player miss over 5 tackles\n", oFvePer);

}

if (filep != NULL)

{

fclose(filep);// closes text file

}// if

}// printTackle

void printMetres(struct node\* headptr)

{

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

float zeroMetres = 0;

float uTenMetres = 0;

float uTwentyMetres = 0;

float oTwentyMetres = 0;

int players = 0;

float zeroPer = 0;

float uTenPer = 0;

float uTwPer = 0;

float oTwPer = 0;

temp = headptr;

FILE\* filep;// creates file to write too

filep = fopen("metreStats.txt", "w");// allows user to write to the text file

while (temp != NULL)

{

if (temp->metresMade == 0) // gives various count of metres covered

{

zeroMetres++;

}

else if (temp->metresMade > 0 && temp->metresMade < 10)

{

uTenMetres++;

}

else if (temp->metresMade >= 10 && temp->metresMade < 20)

{

uTwentyMetres++;

}

else if (temp->metresMade > 20)

{

oTwentyMetres++;

}

temp = temp->NEXT;

}

players = zeroMetres + uTenMetres + uTwentyMetres + oTwentyMetres;// add up all players

zeroPer = (zeroMetres / players) \* 100;

uTenPer = (uTenMetres / players) \* 100; // find percents of metres covered

uTwPer = (uTwentyMetres / players) \* 100;

oTwPer == (oTwentyMetres / players) \* 100;

if (filep != NULL)

{

fprintf(filep, " Metres Stats \n");

fprintf(filep, "===================================================\n");

// prints data to text file

fprintf(filep,"%.2f percent of player make zero metres per match \n", zeroPer);

fprintf(filep,"%.2f percent of player make 0 - 10 metres per match \n", uTenPer);

fprintf(filep,"%.2f percent of player make 10 - 20 metres per match\n", uTwPer);

fprintf(filep,"%.2f percent of player make over 20 metres per match\n", oTwPer);

}

if (filep != NULL)

{

fclose(filep);// closes text file

}// if

}// printMetres

void printToFile(struct node\* headptr)

{

FILE\* filep;// creates file to write too

filep = fopen("Rugby.txt", "w");// allows user to write to the text file

fprintf(filep, " Player Details \n");

fprintf(filep, "==============================================\n");

fprintf(filep, "ID Surname First Name Age Height Weight Club Email Position tackles missed metres made\n");

struct node \*temp;

temp = (struct node\*)malloc(sizeof(struct node));

temp = headptr;

while (temp != NULL)

{

if (filep != NULL)

{

// prints data to text file

fprintf(filep, "%d %s %s %d %.2f %.2f %s %s %s %d %.2f\n", temp->IRFUnum, temp->surname, temp->firstName, temp->age, temp->height, temp->weight,temp->club,temp->email,temp->playerPosition, temp->tacklesMissed, temp->metresMade);

}// if

temp = temp->NEXT;

}

if (filep != NULL)

{

fclose(filep);// closes text file

}// if

}// printToFile

void menu()

{

struct node\* head\_ptr;

int option;

int IRFUnum;

int searchOption;

int updateOption;

int statsOption;

char searchName[30];

int searchID = 0;

int i = 0;

head\_ptr = NULL;

/\*struct node \*temp;

temp = (struct node \*)malloc(sizeof(struct node));

head\_ptr = temp;

FILE \*filep;

filep = fopen("Rugby.txt", "r");

while (!feof(filep))

{

fscanf(filep, "%d", &temp->IRFUnum);

fscanf(filep, "%s", temp->firstName);

fscanf(filep, "%s", temp->surname);

fscanf(filep, "%d", &temp->age);

fscanf(filep, "%.2f", &temp->height);

fscanf(filep, "%.2f", &temp->weight);

fscanf(filep, "%s", temp->club);

fscanf(filep, "%s", temp->email);

fscanf(filep, "%s", temp->playerPosition);

fscanf(filep, "%.2f", &temp->tacklesMissed);

fscanf(filep, "%.2f", &temp->metresMade);

temp = temp->NEXT;

}\*/

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

printf("\*\*\*\*\*\*\*\*\*\*\* Rugby Performance Metric Ltd. \*\*\*\*\*\*\*\*\*\*\*\n");

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

printf("| 1 to add a player | 2 to display all players |\n");

printf("| 3 to search a player | 4 to update a player |\n");

printf("| 5 to delete a player | 6 to generate statistics |\n");

printf("| 7 to list by name | |\n");

printf("| -1 to exit & print detail and stats to file |\n");

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

printf(" Enter your option : ");

scanf("%d", &option);

while (option != -1)// keeps showing menu until user enters -1

{

switch (option)

{

case 1:

if (head\_ptr == NULL)

{

addPlayer\_atStart(&head\_ptr);

}

else

{

printf("Please enter ID : ");

scanf("%d", &searchID);

if (search(head\_ptr, searchID) == 0)

{

printf(" UNUSED ID PLEASE CONTINUE \n");

printf(" THIS ID WILL NOW BE ASSIGNED TO THIS PLAYER\n");

addPlayer(head\_ptr, searchID);

}

else

{

printf("THERE ID IS ALREADY USED\n");

printf("PLEASE TRY AGAIN\n");

}

}

break;

case 2:

display(head\_ptr);

break;

case 3:

printf("Enter 1 to search by ID\n");

printf("Enter 2 to search by Surname\n");

printf(" Enter your option : ");

scanf("%d", &searchOption);

if (searchOption == 1)

{

printf(" Enter players IRFU No : ");

scanf("%d", &searchID);

if (search(head\_ptr, searchID) == 0)

{

printf(" INVALID ID \n");

printf(" PLEASE TRY AGAIN \n");

}

else

{

searchPlayerNo(head\_ptr, searchID);

}

}

else if (searchOption == 2)

{

printf(" Enter players Surname : ");

scanf("%s", searchName);

searchPlayerName(head\_ptr, searchName);

}

break;

case 4:

//printf("Enter 1 to update by ID\n");

//printf("Enter 2 to update by Surname\n");

//printf(" Enter your option : ");

//scanf("%d", &updateOption);

//if (updateOption == 1)

//{

printf(" Enter players IRFU No : ");

scanf("%d", &searchID);

if (search(head\_ptr, searchID) == 0)

{

printf(" INVALID ID \n");

printf(" PLEASE TRY AGAIN \n");

}

else

{

updateByNum(head\_ptr, searchID);

}

//}

//else if (updateOption == 2)

//{

//printf(" Enter players IRFU Surname : ");

//scanf("%s", searchName);

//updateByName(head\_ptr, searchName);

//}

break;

case 5:

printf(" Enter players IRFU No : ");

scanf("%d", &searchID);

//if (head\_ptr == NULL)

//{

// printf(" LIST IS ALREADY EMPTY \n ");

//}

//else if (length(head\_ptr) == 1)

//{

//deletePlayer\_atStart(&head\_ptr, searchID);

//}

//else

//{

if (search(head\_ptr, searchID) !=0)

{

deletePlayer(head\_ptr, searchID);

}

//}

break;

case 6:

printf("Enter 1 to see tackle stats\n");

printf("Enter 2 to see metres stats\n");

printf(" Enter your option : ");

scanf("%d", &statsOption);

if (statsOption ==1)

{

tackleStats(head\_ptr);

}

else if (statsOption == 2)

{

metresStats(head\_ptr);

}

break;

case 7:

displayNames(head\_ptr);

break;

default:

printf("INVALID OPTION, PLEASE CHOOSE AGAIN");

break;

}// switch

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

printf("\*\*\*\*\*\*\*\*\*\*\* Rugby Performance Metric Ltd. \*\*\*\*\*\*\*\*\*\*\*\n");

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

printf("| 1 to add a player | 2 to display all players |\n");

printf("| 3 to search a player | 4 to update a player |\n");

printf("| 5 to delete a player | 6 to generate statistics |\n");

printf("| 7 to list by name | |\n");

printf("| -1 to exit & print detail and stats to file |\n");

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

printf(" Enter your option : ");

scanf("%d", &option);

}// while

printToFile(head\_ptr);// prints player details to file

printTackle(head\_ptr);// prints tackle stats to file

printMetres(head\_ptr);// prints metres stats to file

}