

| **TITLE:**  Array of Structures. |
| --- |

**AIM:** Program to declare an array of structure `players` having data members (name, total matches played, best bowling figure). Program should do the following operations using functions.

1. **Insert Minimum 5 player data in array of structure**
2. **Sort and display this data in descending order of their best bowling figure (if wickets are same then consider less run conceded as priority) and in proper tabular form**
3. **Delete the data for any one player.**
4. **Search for a particular player using its name.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.
4. [**http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**](http://cse.iitkgp.ac.in/~rkumar/pds-vlab/)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Definition:**

Create an array of structure ‘players’ which store information about multiple players having different data members such as name, total matches played, best bowling figure.

Program should read choice from the user and perform following function:

Choice 1: Insert data in array of structure.

Choice 2: Sort and Display

Choice 3: Delete a player

Choice 4: Traverse and search a player with given name.

**Algorithm:**

1) Make the structure.

2) Initialize the functions.

3) Make the main function.

4) Use switch case to show menu driven output.

5) Make an insert function

a. Take the input for each of the players. Store it in the structured array.

6) Make the Sort function.

a. Use bubble sort to sort the array. Sort in descending order with respect to the name ,total matches played and bowling figure.

7) Make the display function

a. Print the array

8) Make the delete function

a. Take the name of the person whose data you want to delete. Then print all the array except that player’s data.

9) Make the Search function

a. Take the name of the person whose data you want to search. Compare this name with every other name in the list. If there is a match then print the data for the player.

**Implementation details:**

//Meet Gala 16010121051 A3 Expt 6 //

#include<stdio.h>

#include<string.h>

struct players

{

char name[20];

int matchPlayed;

int bowFig;

};

struct players arr[5];

int search(struct players \*arr,char name[20],int n)

{

for(int i=0;i<n;i++)

if(strcmp(arr[i].name,name)==0)

return i;

return -1;

}

void sort(struct players \*arr)

{

for(int i=0;i<5;i++)

{

int max=i;

for(int j=i;j<5;j++)

{

if(arr[max].bowFig<=arr[j].bowFig){

if(arr[max].bowFig==arr[j].bowFig)

max=(arr[max].matchPlayed>arr[j].matchPlayed)?max:j;

else

max=j;

}

}

struct players temp=arr[i];

arr[i]=arr[max];

arr[max]=temp;

}

}

int main()

{

printf("Enter the Data\n");

for(int i=0;i<5;i++)

{

printf("Enter Player name: ");

scanf("%s",arr[i].name);

printf("Enter %s's Matches played: ",arr[i].name);

scanf("%d",&arr[i].matchPlayed);

printf("Enter %s's Bowling Figure: ",arr[i].name);

scanf("%d",&arr[i].bowFig);

}

sort(arr);

printf("\nName\tMatches\tBowling fig.\n\n");

for(int i=0;i<5;i++)

printf("%s\t%d\t%d\n",arr[i].name,arr[i].matchPlayed,arr[i].bowFig);

printf("\n Enter which player you want to delete: ");

char s[10];

scanf("%s",s);

int n=search(arr,s,5);

if(n!=-1)

while(n++<5)

arr[n-1]=arr[n];

else

printf("Player to be deleted not found!\n");

printf("\nAfter Deleting\n");

printf("\nName\tMatches\tBowling fig.\n\n");

for(int i=0;i<4;i++)

printf("%s\t%d\t%d\n",arr[i].name,arr[i].matchPlayed,arr[i].bowFig);

printf("\nEnter the player you want to search: ");

char name[20];

scanf("%s",name);

int ans=search(arr,name,4);

if(ans!=-1)

printf("%s\t%d\t%d\n",arr[ans].name,arr[ans].matchPlayed,arr[ans].bowFig);

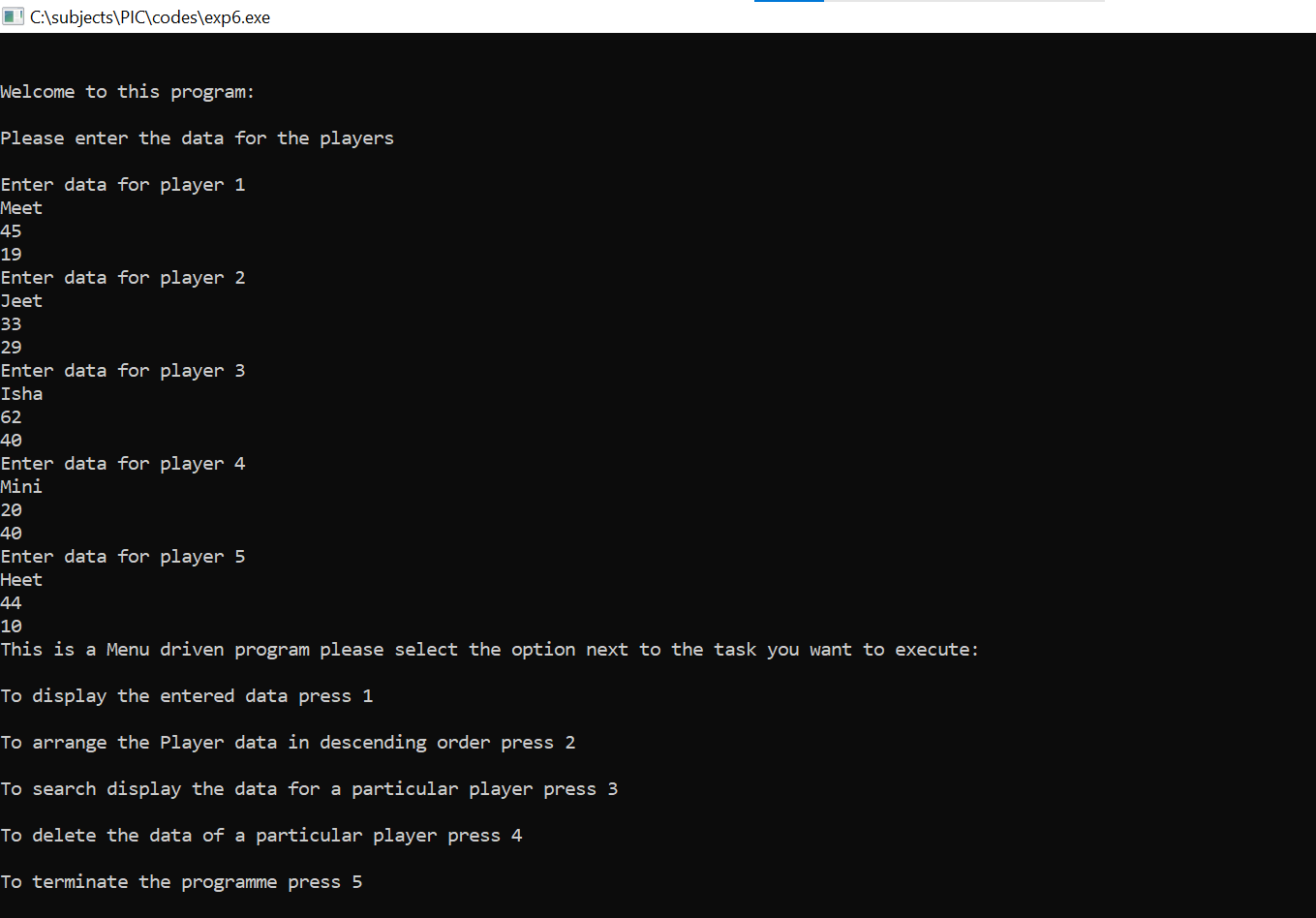
else

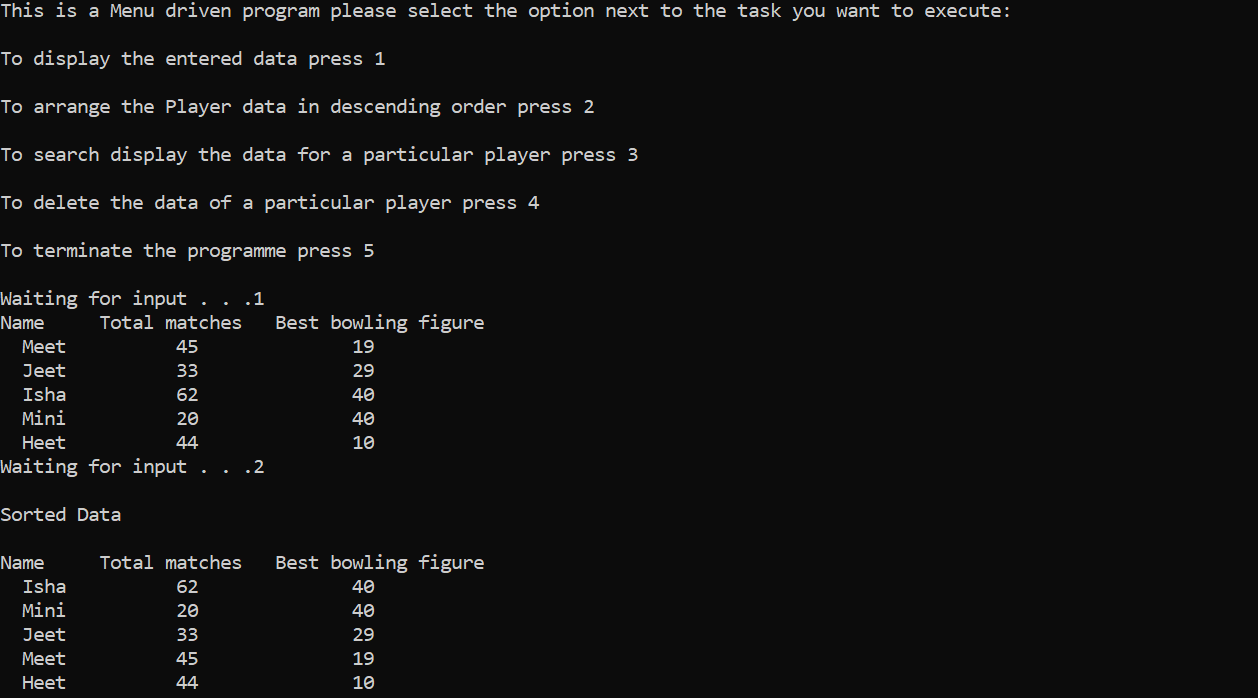
printf("Player Not Found!\n");

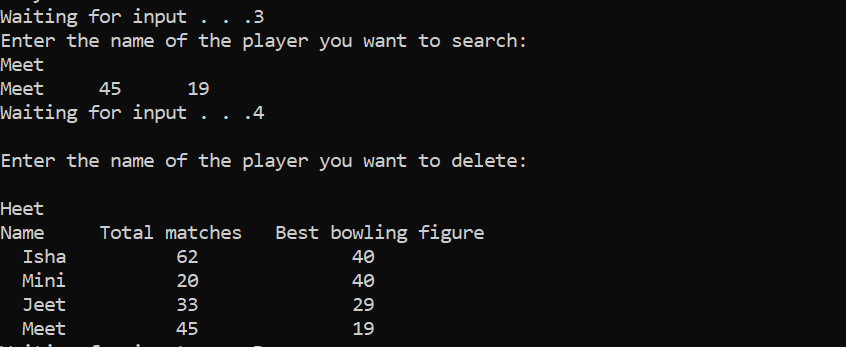
return 0;

}

**Output(s):**

****

****

****

**Conclusion:**

We have thus written a program which satisfies the problem statement and gives a menu driven output; and allows for the insertion of data of minimum 5 players; deletes the data of one player; sorts and displays the list in descending order; and searches the array for the data of a particular person.

**Post Lab Descriptive Questions**

1. **Comment on the output of the following C code.**

#include <stdio.h>

struct temp

{

int a;

int b;

int c;

};

main()

{

struct temp p[] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

}

**Ans**. The output will be nothing. However, an array of a structure with the name temp will be created which will consist of three integers. The first array will be (1, 2, 3), the second array will be (4, 5, 6), and the third array will be (7, 8, 9) which will be initialized to variables (a, b, c) respectively.

1. **Consider the following C code. What will be the output?**

#include<stdio.h>

struct st

{

int x;

struct st next;

};

int main()

{

struct st temp;

temp.x = 10;

temp.next = temp;

printf("%d", temp.next.x);

return 0;

}

(A) Compiler Error

(B) 10

(C) Runtime Error

(D) Garbage Value

**Ans.** (A) Compiler Error

1. **Difference between Structure and Union.**

| **Structure** | **Union** |
| --- | --- |
| A structure is a user-defined data type available in C that allows to combining data items of different kinds. Structures are used to represent a record. | A union is a special data type available in C that allows storing different data types in the same memory location. |

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**