Ketaki Chhayendra Mahajan

**Batch:** P1-2

**Roll Number:** 16014022050

**Experiment / assignment / tutorial no.:** 6

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **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:**

The objective of this program is to create a menu-driven program in which user is expected to enter a choice to carry out the following functions -

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

Upon entering the choice, the respective function should be carried out and displayed by calling it in the main function. The output of each function (if required) should be displayed in a proper tabular format.

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

**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. The 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. Create the main function.
4. Create a do while loop.
5. Use switch case inside the loop and create required cases to show menu driven output.
6. Make an insert function
   1. Input data for each players and store it in the structured array.
7. Make the Sort function.
   1. Use bubble sort to sort the array in descending order with respect to the name, total matches played and bowling figure.
8. Make the Display function
   1. Print the array
9. Make the Delete function
10. Take the name of the person whose record you want to delete and then print the array except that particulars player’s data. If there is no match, the print ‘NO RECORD FOUND’.
11. Make the Search function
    1. Take the name of the person whose record 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, else print ‘NO RECORD FOUND’.
12. If choice entered is equal to case, call respective function and display output.
    1. If choice entered is not present in cases implemented, print default case, ‘INVALID INPUT’.

**Implementation details:**

// experiment 6: Ketaki Mahajan, 16014022050, P1-2

#include<stdio.h>

#include<string.h>

struct players

{

char name[20];

int matches\_played;

float bow\_fig;

};

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].bow\_fig <= arr[j].bow\_fig)

{

if(arr[max].bow\_fig == arr[j].bow\_fig)

max = (arr[max].matches\_played > arr[j].matches\_played) ? max:j;

else

max = j;

}

}

struct players temp = arr[i];

arr[i] = arr[max];

arr[max] = temp;

}

}

int main()

{

int choice;

printf(" PIC Experiment 6 - 16014022050 \n");

do

{

printf("\n\n 1. Insert data in array of structure \n 2. Sort and Display \n 3. Delete a player \n 4. Traverse and search a player with given name \n 5. Exit");

printf("\n\n Enter a choice: ");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf(" You have chosen option 1.");

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

{

printf("\n Enter player name: ");

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

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

scanf("%d", &arr[i].matches\_played);

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

scanf("%f", &arr[i].bow\_fig);

}

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("\n\n Records entered are: ");

printf("\n\n Name \t\t Matches Played \t Best Bowling Figure\n\n");

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

printf(" %s \t\t %d \t\t\t %.2f\n", arr[i].name, arr[i].matches\_played, arr[i].bow\_fig);

printf("\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case 2:

printf(" You have chosen option 2.");

printf("\n\n Records sorted in descending order of best bowling figure is: ");

sort(arr);

printf("\n\n Name \t\t Matches Played \t Best Bowling Figure\n\n");

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

printf(" %s \t\t %d \t\t\t %.2f\n", arr[i].name, arr[i].matches\_played, arr[i].bow\_fig);

printf("\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case 3:

printf(" You have chosen option 3.");

printf("\n\n Enter name of the player you wish 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 NOT FOUND\n");

printf("\n Updated records after deletion is: \n");

printf("\n Name \t\t Matches Played \t Best Bowling Figure\n\n");

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

{

printf(" %s \t\t %d \t\t\t %.2f\n", arr[i].name, arr[i].matches\_played, arr[i].bow\_fig);

}

printf("\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case 4:

printf(" You have chosen option 4.");

printf("\n\n Enter name of player you wish to search: ");

char name[20];

scanf("%s", name);

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

if(ans != -1)

printf(" %s \t\t %d \t\t\t %.2f\n", arr[ans].name, arr[ans].matches\_played, arr[ans].bow\_fig);

else

printf(" PLAYER NOT FOUND\n");

printf("\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

case 5:

printf(" You have exited the program.");

exit(0);

default:

printf(" INVALID INPUT ");

printf("\n Enter number from 1 to 4 ");

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

}

while(choice != 5);

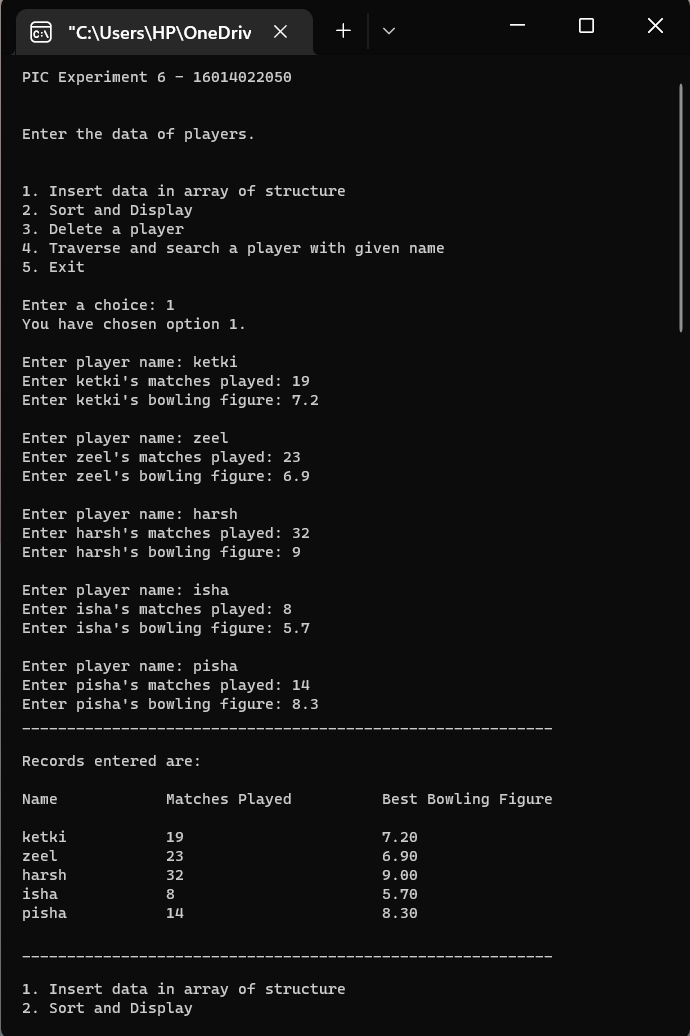
printf("\n\n End of program.");

return 0;

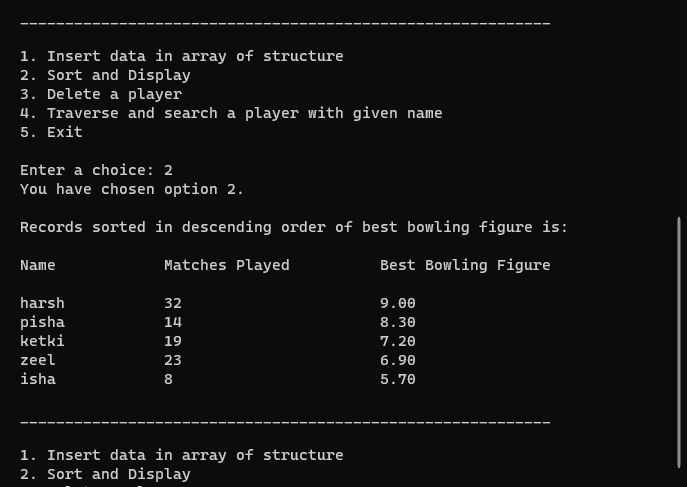
}

**Output(s):**

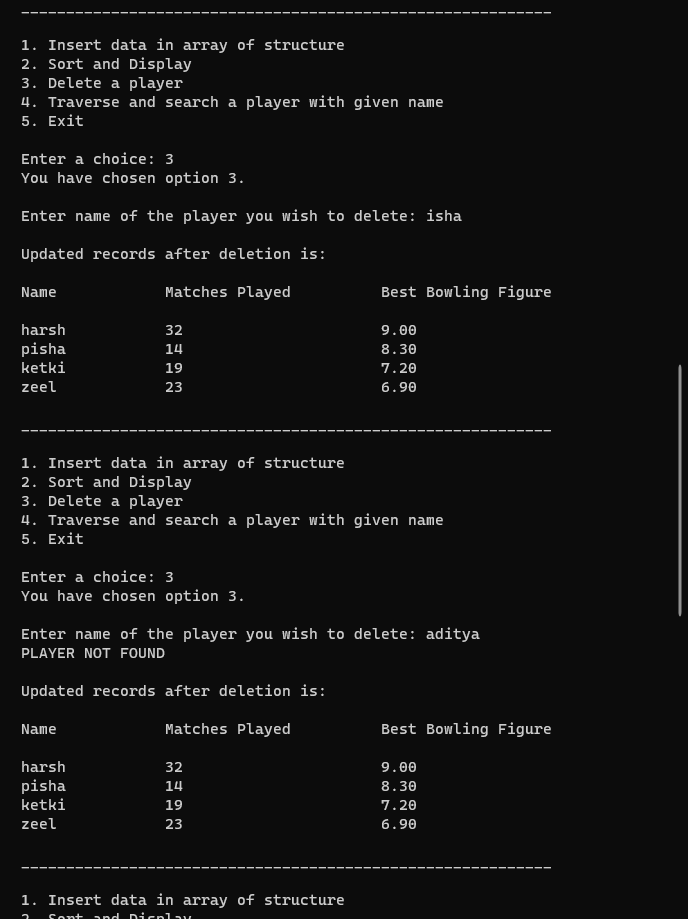
Choice 1 –



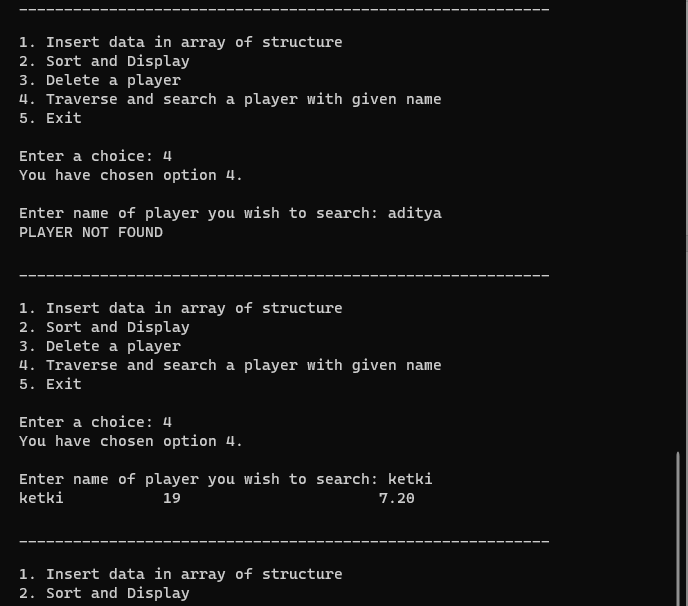
Choice 2 –



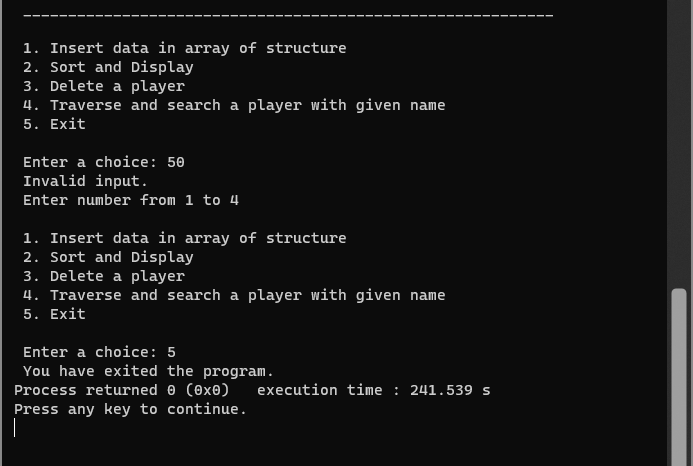
Choice 3 –



Choice 4 –

****

Choice 5 (exit) and invalid choice –



**Conclusion:**

To conclude, carrying out experiment 5 has resulted in learning various new concepts such as–

* creating functions and calling them in the main function.
* creating structures and implementing them aptly.
* inputting data in the array of structure.
* traversing arrays.
* deleting and searching records in array.
* using *#include<string.h>* header file to declares set of functions of strings such as *strcmp* to compare two strings as used in above code.

Hence, I was able to successfully carry out experiment 6 and fulfil the objectives that the problem defined as seen in the output.

**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}};

}

The output will be nothing as the *printf* statement has not been used. 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?
2. Compiler Error
3. 10
4. Runtime Error
5. Garbage Value

#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;

}

The output is **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**