Batch: C2-1 **Roll No.: 16010122104**

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.

a. **Insert Minimum 5 player data in array of structure**
b. **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**
c. **Delete the data for any one player.**
d. **Search for a particular player using its name.**

Expected OUTCOME of Experiment:

To declare an array of structure `players` have different data members, and sort and delete as per the user's requirement.

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/**

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:

Implementation details:

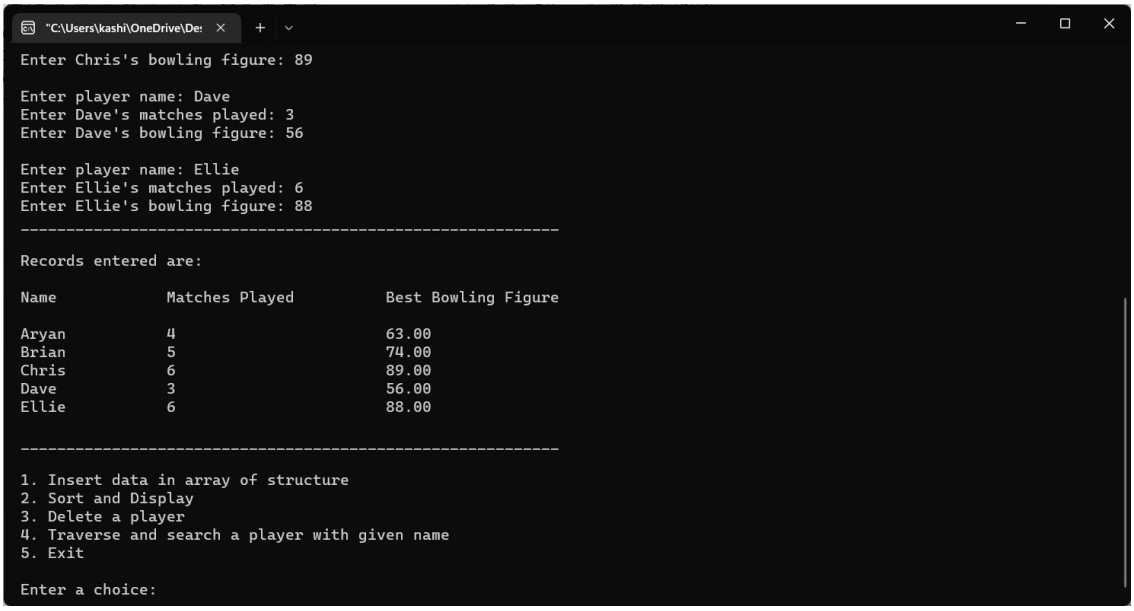
```
#include<stdio.h>
#include<string.h>

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("\n\n Enter the data of players. \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. \n");
            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
```

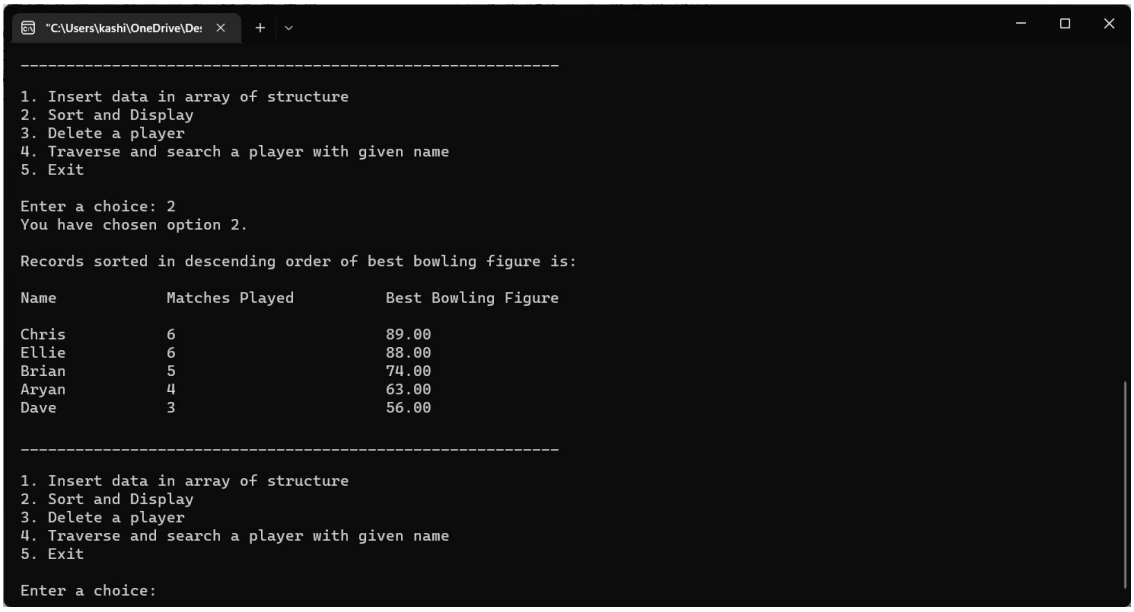
```
_____");
            break;


        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;
}
```

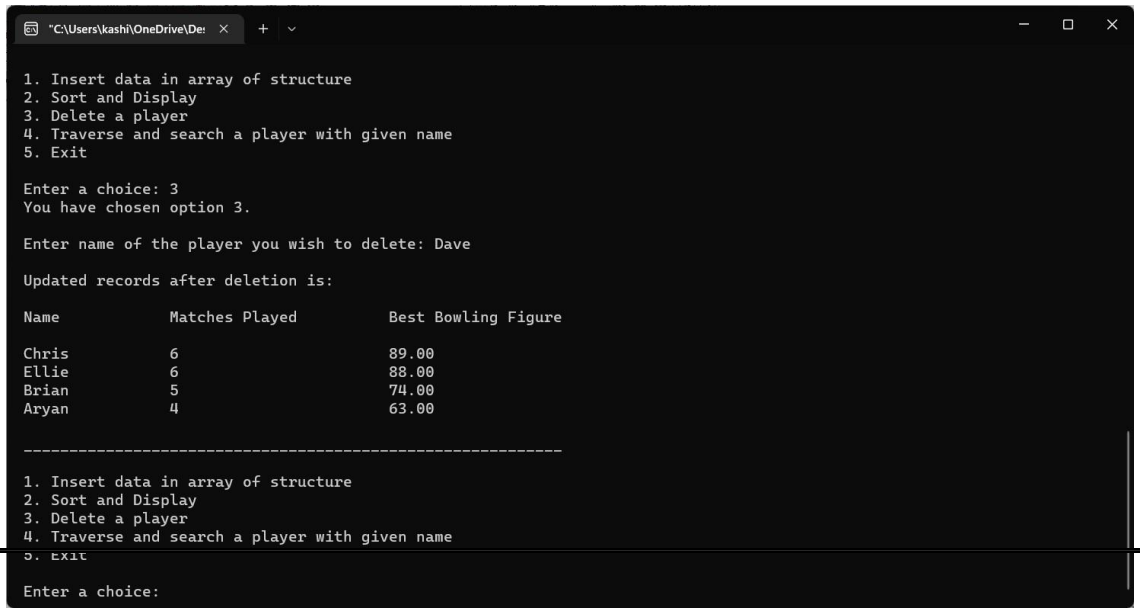
Output(s):



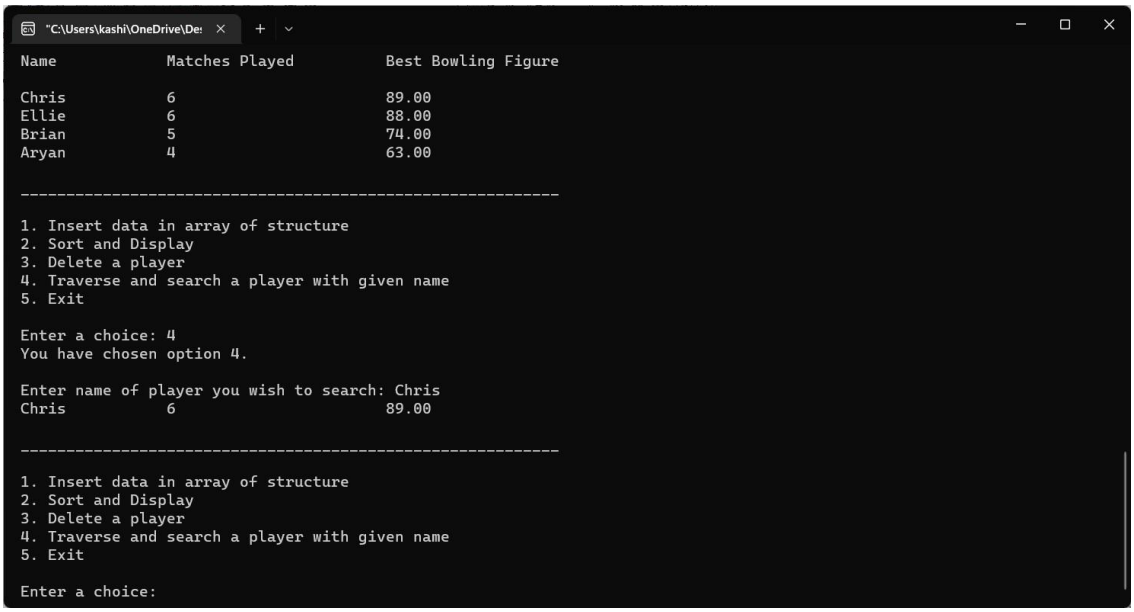
```
"C:\Users\kashi\OneDrive\De:
Enter Chris's bowling figure: 89

Enter player name: Dave
Enter Dave's matches played: 3
Enter Dave's bowling figure: 56

Enter player name: Ellie
Enter Ellie's matches played: 6
Enter Ellie's bowling figure: 88
------------------------------------------------------------

Records entered are:

Name            Matches Played          Best Bowling Figure

Aryan           4                       63.00
Brian           5                       74.00
Chris           6                       89.00
Dave            3                       56.00
Ellie           6                       88.00

------------------------------------------------------------

1. Insert data in array of structure
2. Sort and Display
3. Delete a player
4. Traverse and search a player with given name
5. Exit

Enter a choice:
```

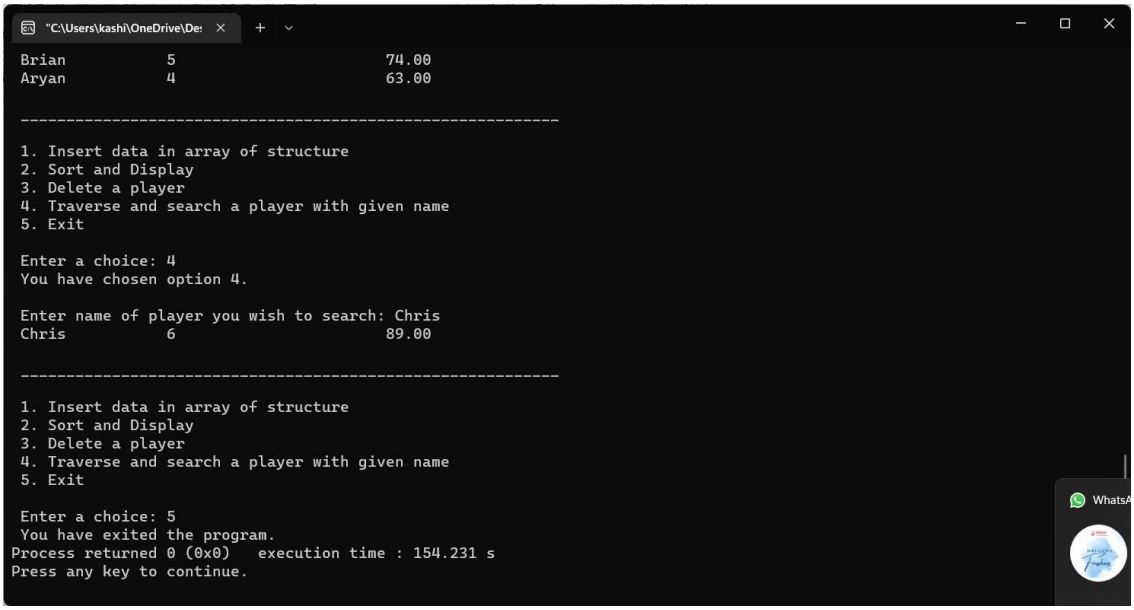


```
"C:\Users\kashi\OneDrive\De:
------------------------------------------------------------

1. Insert data in array of structure
2. Sort and Display
3. Delete a player
4. Traverse and search a player with given name
5. Exit

Enter a choice: 2
You have chosen option 2.

Records sorted in descending order of best bowling figure is:

Name            Matches Played          Best Bowling Figure

Chris           6                       89.00
Ellie           6                       88.00
Brian           5                       74.00
Aryan           4                       63.00
Dave            3                       56.00

------------------------------------------------------------

1. Insert data in array of structure
2. Sort and Display
3. Delete a player
4. Traverse and search a player with given name
5. Exit

Enter a choice:
```



```
"C:\Users\kashi\OneDrive\De:

1. Insert data in array of structure
2. Sort and Display
3. Delete a player
4. Traverse and search a player with given name
5. Exit

Enter a choice: 3
You have chosen option 3.

Enter name of the player you wish to delete: Dave

Updated records after deletion is:

Name            Matches Played          Best Bowling Figure

Chris           6                       89.00
Ellie           6                       88.00
Brian           5                       74.00
Aryan           4                       63.00

------------------------------------------------------------

1. Insert data in array of structure
2. Sort and Display
3. Delete a player
4. Traverse and search a player with given name
5. Exit

Enter a choice:
```





Conclusion:

We have successfully declared an array of structure `players` having different data members through which we can sort and delete particular data as per the user's requirement.

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}};
   }
```

Answer:

The above code just returns a blank page with no output as we have not initiated any operation.

2. **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

Answer:
The output for the above code will be a compiler error (option A), the field next has an incomplete type.

3. **Difference between Structure and Union.**

Answer:
Structures in C is a user-defined data type available in C that allows to combining of data items of different kinds. Structures are used to represent a record.

Union in C is a special data type available in C that allows storing different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple purposes.

Date: _____ **Signature of faculty in-charge**