

**K. J. Somaiya College of Engineering, Mumbai-77**

(A Constituent College of Somaiya Vidyavihar University)

**Batch: G3**

**Roll No.: 16010421063**

**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:**

CO4: Design modular programs using functions and demonstrate the concept of pointers and file handling.

CO3: Illustrate the use of derived and structured data types such as arrays, strings, structures and unions

---

**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 structured '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 an array of structure.

Choice 2: Sort and Display

Choice 3: Delete a player

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

**Algorithm:**

1. Create a structure which contains names, matches played, wickets and runs.
2. Get the value from the user about how many player names the user wants to input.
3. Make an array of structure of that size.
4. Take values such as names, matches played, wickets and runs by traversing through the structure and assign the value given by the user
5. Give user a choice-
  - a. Sort the players according to wickets
  - b. Delete one of the player data
  - c. Search for the data of one of the players
  - d. Exit the program
6. If user chooses sort , then by using bubble sort we will sort the array and display the data
7. If the user chooses delete, then ask the user the name of the user you want to delete. Then compare the user entered value and the elements of the array. If both string matches then delete the particular array element
8. If the user chooses to search, then ask the user the name of the user they want to search for. Compare the user entered value and the elements of the array. If both strings match, display the elements of the array.
9. If the user chooses to insert an element we assign the values to the next memory address of array
10. If user chooses Exit, just exit the code by writing "return 0;"

**Implementation details:**

```
#include<stdio.h>

struct players
{
    char name[20];
    int matches_played;
    int wickets;
    int runs;
};

void display(struct players arr[],int n)
//Function to sort and display the elements of the array
{
    printf("\nName\t Matches Played\t Bowling Figure");

    printf("\n-----");
};

    struct players temp;
    for (int i=0;i<n;i++)
    {
        for (int j=i+1;j<n;j++)
        {
            if (arr[i].wickets<arr[j].wickets)
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
            if (arr[i].wickets==arr[j].wickets)
                //If wickets are equal we compare them by
using less runs conceded
            {
                if (arr[i].runs>arr[j].runs)
                {
```

```
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    for (int i=0;i<n;i++)
    {
        printf("\n%s\t\t %d\t\t %d/%d",arr[i].name,arr[i].matches_played,arr[i].wickets,arr[i].runs);
    }

    printf("\n-----
-");

    return arr;
}

int add_player(struct players arr[],int n)
//function to add array to the next available part of the array
{
    int x;
    printf("Number of players you want to add: ");
    scanf("%d",&x);
    for (int i=n;i<n+x;i++)
    {
        printf("Enter name of player: ");
        scanf("%s",&arr[i].name);
        printf("Enter matches played: ");
        scanf("%d",&arr[i].matches_played);
        printf("Enter wickets and runs conceded: ");
        scanf("%d%d",&arr[i].wickets,&arr[i].runs);
        printf("\n");
    }
    n=n+x;
    display(arr,n);
    return (n);
}
```

```
}

int del(struct players arr[],int n)
//Function for deleting an element from the array
{
    char name1[20];
    printf("\nEnter the name you want to Delete for: ");
    scanf("%s",name1);
    for (int i=0;i<n;i++)
    {

        if (strcmp(arr[i].name,name1)==0)
        {
            for (int j=i;j<n;j++)
            {
                arr[j]=arr[j+1];
            }
            n=n-1;
        }
    }
    display(arr,n);
    return n;
}

void search_player(struct players arr[],int n)
//Function to search an element of the array
{
    char name1[20];
    printf("\nEnter the name you want to search for: ");
    scanf("%s",name1);
    printf("\nName\t Matches Played\t Bowling Figure");

    printf("\n-----");
    ");
    for (int i=0;i<n;i++)
    {

        if (strcmp(arr[i].name,name1)==0)
        {
```

Department of Science and Humanities

```
printf("\n3. Search for a player using name");
printf("\n4. Insert players");
printf("\n5. Exit the program");
printf("\nEnter your choice: ");
scanf("%d",&choice);
switch(choice)
{
case (1):
{
    display(arr,n);
    break;
}
case (2):
{
    n=del(arr,n);
    break;
}
case (3):
{
    search_player(arr,n);
    break;
}
case (4):
{
    n=add_player(arr,n);
    break;
}

case (5):
    return 0;
}

}while(choice!=5);
}
```

**Output(s):**

**K. J. Somaiya College of Engineering, Mumbai-77**

(A Constituent College of Somaiya Vidyavihar University)



## K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

 D:\College\PIC\EXP6\insert.exe

```
Enter number of players(n>=5): 5
Enter data:
Enter name of player: Player1
Enter matches played: 5
Enter wickets and runs conceded: 20 40
```

```
Enter name of player: Player2
Enter matches played: 8
Enter wickets and runs conceded: 20 30
```

```
Enter name of player: Player3
Enter matches played: 9
Enter wickets and runs conceded: 40 20
```

```
Enter name of player: Player4
Enter matches played: 4
Enter wickets and runs conceded: 10 30
```

```
Enter name of player: Player5
Enter matches played: 7
Enter wickets and runs conceded: 30 10
```

```
1. Sort and display player data
2. Delete data for any one player
3. Search for a player using name
4. Insert players
5. Exit the program
Enter your choice: 1
```

Name	Matches Played	Bowling Figure
Player3	9	40/20
Player5	7	30/10
Player2	8	20/30
Player1	5	20/40
Player4	4	10/30

```
1. Sort and display player data
2. Delete data for any one player
3. Search for a player using name
4. Insert players
5. Exit the program
Enter your choice: 4
```

```
Number of players you want to add: 1
Enter name of player: Arya
Enter matches played: 9
Enter wickets and runs conceded: 50 20
```

## K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

D:\College\PIC\EXP6\insert.exe

Name	Matches Played	Bowling Figure
Arya	9	50/20
Player3	9	40/20
Player5	7	30/10
Player2	8	20/30
Player1	5	20/40
Player4	4	10/30

1. Sort and display player data  
2. Delete data for any one player  
3. Search for a player using name  
4. Insert players  
5. Exit the program  
Enter your choice: 2

Enter the name you want to Delete for: Arya

Name	Matches Played	Bowling Figure
Player3	9	40/20
Player5	7	30/10
Player2	8	20/30
Player1	5	20/40
Player4	4	10/30

1. Sort and display player data  
2. Delete data for any one player  
3. Search for a player using name  
4. Insert players  
5. Exit the program  
Enter your choice: 3

Enter the name you want to search for: Player3

Name	Matches Played	Bowling Figure
Player3	9	40

1. Sort and display player data  
2. Delete data for any one player  
3. Search for a player using name  
4. Insert players  
5. Exit the program  
Enter your choice: 5

Process returned 0 (0x0) execution time : 107.122 s  
Press any key to continue.

### Conclusion:

Department of Science and Humanities

**We successfully wrote a program which used an array of structures. Also implemented sorting, searching and deleting algorithm on that**

### **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}};
}
```

**And- There will be no output as there is no output function (printf(), putchar(), puts()).**

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

**K. J. Somaiya College of Engineering, Mumbai-77**

(A Constituent College of Somaiya Vidyavihar University)

- (B) 10
- (C) Runtime Error
- (D) Garbage Value

**Ans- A)Compiler Error**

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

<b>Structure</b>	<b>Union</b>
The keyword Struct is used to define a structure	The keyword Union is used to define a union
When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is greater than or equal to the sum of sizes of its members	when a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is equal to the size of largest member
Each member within a structure is assigned unique storage area of location.	Memory allocated is shared by individual members of union.
Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter other member values.
Individual members can be accessed at a time.	Only one member can be accessed at a time.
Several members of a structure can initialize at once.	Only the first member of a union can be initialized.

**Date:** \_\_\_\_\_

**Signature of faculty in-charge**