Batch: P5-2	**Roll No.: 16010422185**
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:

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:

Step 1: START.

Step 2: Enter choice to perform various operations -

1. Insert data in an array of structure.

2. Sort and Display.

- Using algorithms to sort data based on runs and wickets of players

3. Delete a player:

- remove player from structure and print updated list.

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

- display information of the player which is being searched.

5. Exit.

Step 3:STOP

Implementation details:

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

struct player
{
   char name[50];
   int matches, wickets, runs;
};

void getData(struct player[]);
void sort(struct player[], int);
void display(struct player[], int);
int search(struct player[], int);
void deletePlayer(struct player[], int);

int main()
{
   struct player p[5];
   int n = 5, choice;

   do
   {
      printf("\n\n1. Insert data in array of structure.");
      printf("\n2. Sort and Display.");
      printf("\n3. Delete a player.");
      printf("\n4. Traverse and search a player with given name.");
      printf("\n5. Exit.");
      printf("\n\nEnter choice: ");
      scanf("%d", &choice);

      switch (choice)
      {
      case 1:
         getData(p);
         break;

      case 2:
         sort(p, 5);
         break;

      case 3:
         deletePlayer(p, 5);
         break;
```

```
      case 4:
         search(p, 5);
         break;

      default:
         printf("Exit");
         break;
      }
   } while (choice < 5);

   return 0;
}

void getData(struct player p[5])
{
   int n = 5;

   for (int i = 0; i < n; i++)
   {
      printf("\n---Player %d---", i + 1);
      printf("\nEnter player's name: ");
      fflush(stdin);
      gets(p[i].name);
      printf("Enter no. of matches played: ");
      scanf("%d", &p[i].matches);
      printf("Enter wickets: ");
      scanf("%d", &p[i].wickets);
      printf("Enter runs: ");
      scanf("%d", &p[i].runs);
   }
}

void sort(struct player p[5], int n)
{
   struct player temp;

   for (int i = 0; i < n - 1; i++)
   {
      for (int j = 0; j < n - 1; j++)
      {
         if (p[j].wickets < p[j + 1].wickets)
         {
            temp = p[j];
            p[j] = p[j + 1];
            p[j + 1] = temp;
         }
      }
```

```
    }

    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - 1; j++)
        {
            if (p[j].wickets == p[j + 1].wickets)
            {
                if (p[j].runs > p[j + 1].runs)
                {
                    temp = p[j];
                    p[j] = p[j + 1];
                    p[j + 1] = temp;
                }
            }
        }
    }

    printf("Sorted list is:");
    display(p, n);
}

int search(struct player p[5], int n)
{
    int i;
    char playerName[50];
    printf("\nEnter name of the player you want to search: ");
    fflush(stdin);
    gets(playerName);

    for (int i = 0; i < n; i++)
    {
        if (strcmp(p[i].name, playerName) == 0)
        {
            printf("---Player Found!---");
            printf("\nPlayer Name: %s", p[i].name);
            printf("\nMatches: %d", p[i].matches);
            printf("\nWickets: %d", p[i].wickets);
            printf("\nRuns: %d", p[i].runs);
            break;
        }
    }
    return i;
}

void deletePlayer(struct player p[5], int n)
{
```

```
    int i;
    char playerName[50];
    printf("\n\nEnter name of the player you want to delete: ");
    fflush(stdin);
    gets(playerName);

    for (int i = 0; i < n; i++)
    {
        if (strcmp(p[i].name, playerName) == 0)
        {
            for (int j = i; j < n; j++)
            {
                p[i] = p[i + 1];
            }
            n--;
        }
    }

    printf("\nPlayer deleted");
    display(p, n);
}

void display(struct player p[5], int n)
{
    printf("\n\nPlayers list: ");
    for (int i = 0; i < n; i++)
    {
        printf("\n\n---Player %d---", i + 1);
        printf("\nPlayer Name: %s", p[i].name);
        printf("\nMatches: %d", p[i].matches);
        printf("\nWickets: %d", p[i].wickets);
        printf("\nRuns: %d", p[i].runs);
    }
}
```

Output(s):

```
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 choice: 1

---Player 1---
Enter player's name: Pratham
Enter no. of matches played: 100
Enter wickets: 25
Enter runs: 550

---Player 2---
Enter player's name: om
Enter no. of matches played: 55
Enter wickets: 25
Enter runs: 50

---Player 3---
Enter player's name: rikhav
Enter no. of matches played: 100
Enter wickets: 5
Enter runs: 50

---Player 4---
Enter player's name: raghav
Enter no. of matches played: 105
Enter wickets: 50
Enter runs: 5

---Player 5---
Enter player's name: dhruv
Enter no. of matches played: 60
Enter wickets: 0
Enter runs: 400
```

```
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 choice: 2
Sorted list is:

Players list:

---Player 1---
Player Name: raghav
Matches: 105
Wickets: 50
Runs: 5

---Player 2---
Player Name: om
Matches: 55
Wickets: 25
Runs: 50

---Player 3---
Player Name: Pratham
Matches: 100
Wickets: 25
Runs: 550

---Player 4---
Player Name: rikhav
Matches: 100
Wickets: 5
Runs: 50

---Player 5---
Player Name: dhruv
Matches: 60
Wickets: 0
Runs: 400
```

```
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 choice: 3


Enter name of the player you want to delete: rikhav

Player deleted

Players list:

---Player 1---
Player Name: raghav
Matches: 105
Wickets: 50
Runs: 5

---Player 2---
Player Name: om
Matches: 55
Wickets: 25
Runs: 50

---Player 3---
Player Name: Pratham
Matches: 100
Wickets: 25
Runs: 550

---Player 4---
Player Name: dhruv
Matches: 60
Wickets: 0
Runs: 400
```

```
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 choice: 4

Enter name of the player you want to search: Pratham
---Player Found!---
Player Name: Pratham
Matches: 100
Wickets: 25
Runs: 550

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 choice: 5
Exit
Process returned 0 (0x0)   execution time : 219.906 s
Press any key to continue.
```

Conclusion:

5 Players data is stored using structures and various operations like search,delete and display are performed using proper algorithm.

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:
This Program will generate an error message as no data type is mentioned before main() and no return statement is used. If the mentioned problems are corrected then the program will run and then an array of structure p of size 3 will be generated.

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

Ans:
(C) Runtime Error

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

Difference Between Structure and Union :

Structure	Union
i. Access Members	
We can access all the members of structure at anytime.	Only one member of union can be accessed at anytime.
ii. Memory Allocation	
Memory is allocated for all variables.	Allocates memory for variable which variable require more memory.
iii. Initialization	
All members of structure can be initialized	Only the first member of a union can be initialized.
iv. Keyword	
'struct' keyword is used to declare structure.	'union' keyword is used to declare union.
v. Syntax	

`struct struct_name`
`{`
`  structure element 1;`
`  structure element 2;`
`  ----------`
`  ----------`
`  structure element n;`
`}struct_var_nm;`
	`union union_name`
	`{`
	`  union element 1;`
	`  union element 2;`
	`  ----------`
	`  ----------`
	`  union element n;`
	`}union_var_nm;`

vi. Example	

`struct item_mst`
`{`
`  int rno;`
`  char nm[50];`
`}it;`
	`union item_mst`
	`{`
	`  int rno;`
	`  char nm[50];`
	`}it;`

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