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

(A Constituent College of Somaiya Vidyavihar University)

Name: Chandana Ramesh Galgali
Batch: P3-3 Roll No.: 16010422234
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 an array of structure.**

b. **Sort and display this data in descending order of their best bowling figures (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 their name.**

Expected OUTCOME of Experiment: The program should be able to take information about the 5 players as input, sort them in descending order of their bowling figures and display it in a tabular pattern. The program then should be able to delete the data for any one player as inputted by the user, it should also be able to search for any player according to the user's input.

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/**

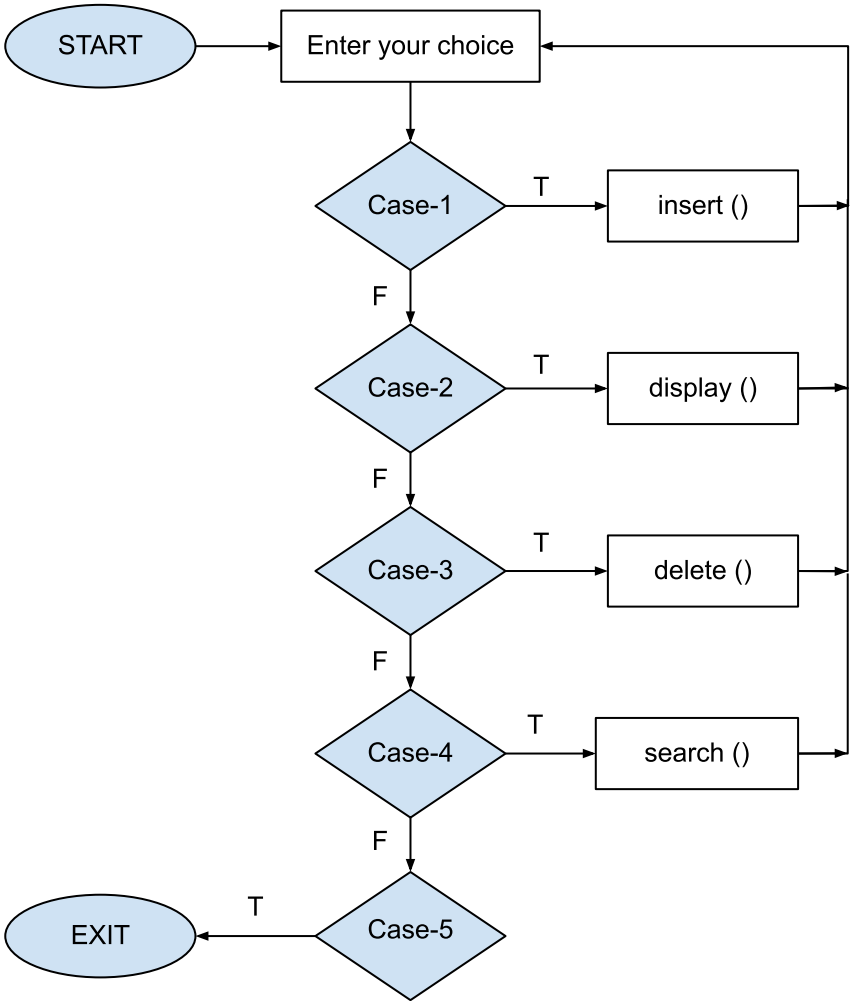
Problem Definition:

Create an array of structures-'players' which can 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 functions:

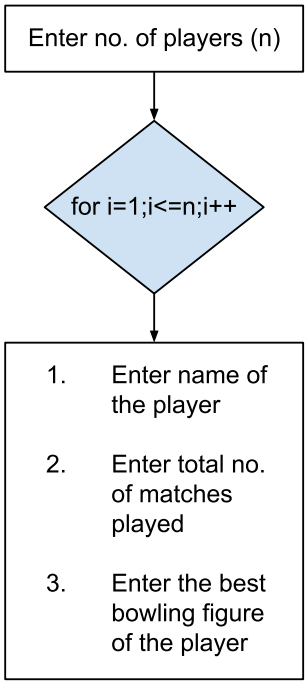
Choice 1: Insert data in the array of structure.
Choice 2: Sort and Display.
Choice 3: Delete a player.
Choice 4: Traverse and search a player with the name given by the user.

Algorithm:

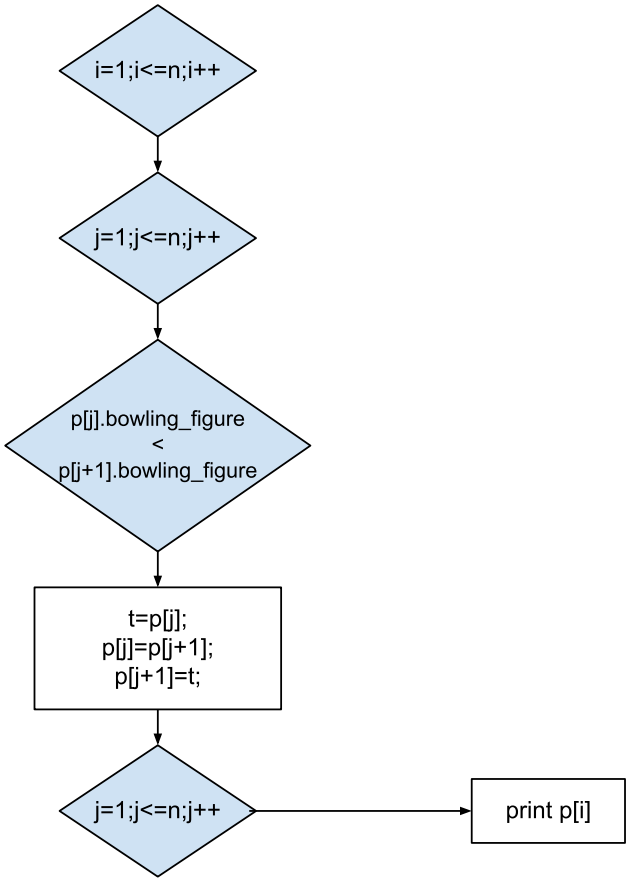
For main() function:



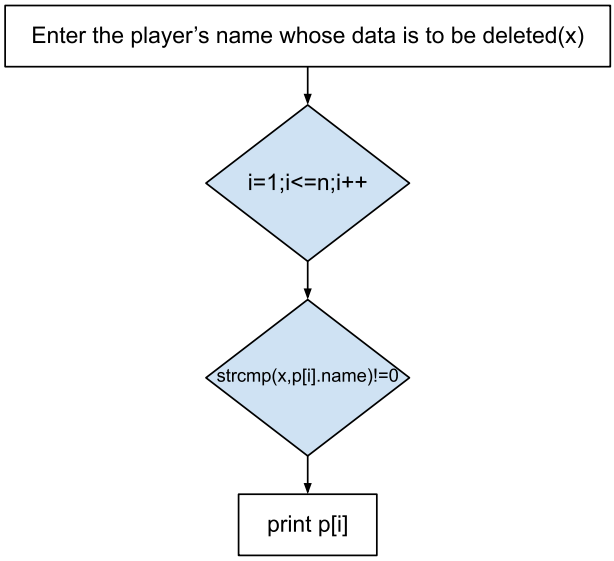
For insert() function:



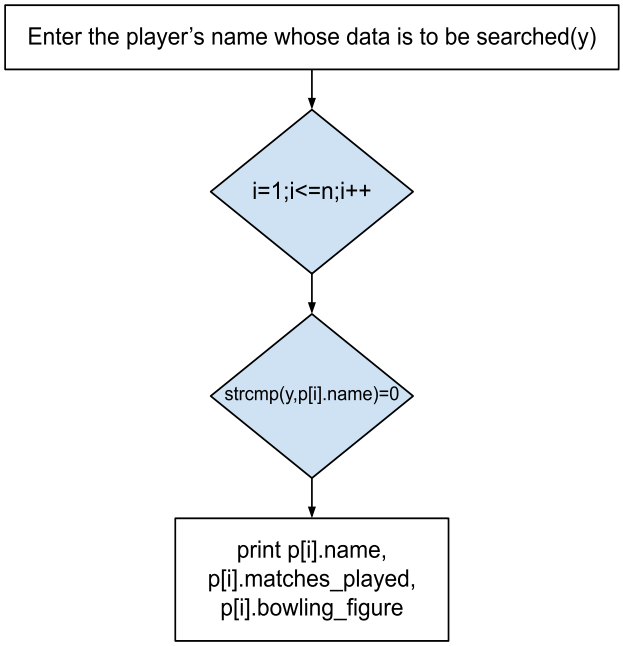
For display() function:



For delete() function:



For search() function:



Implementation details:

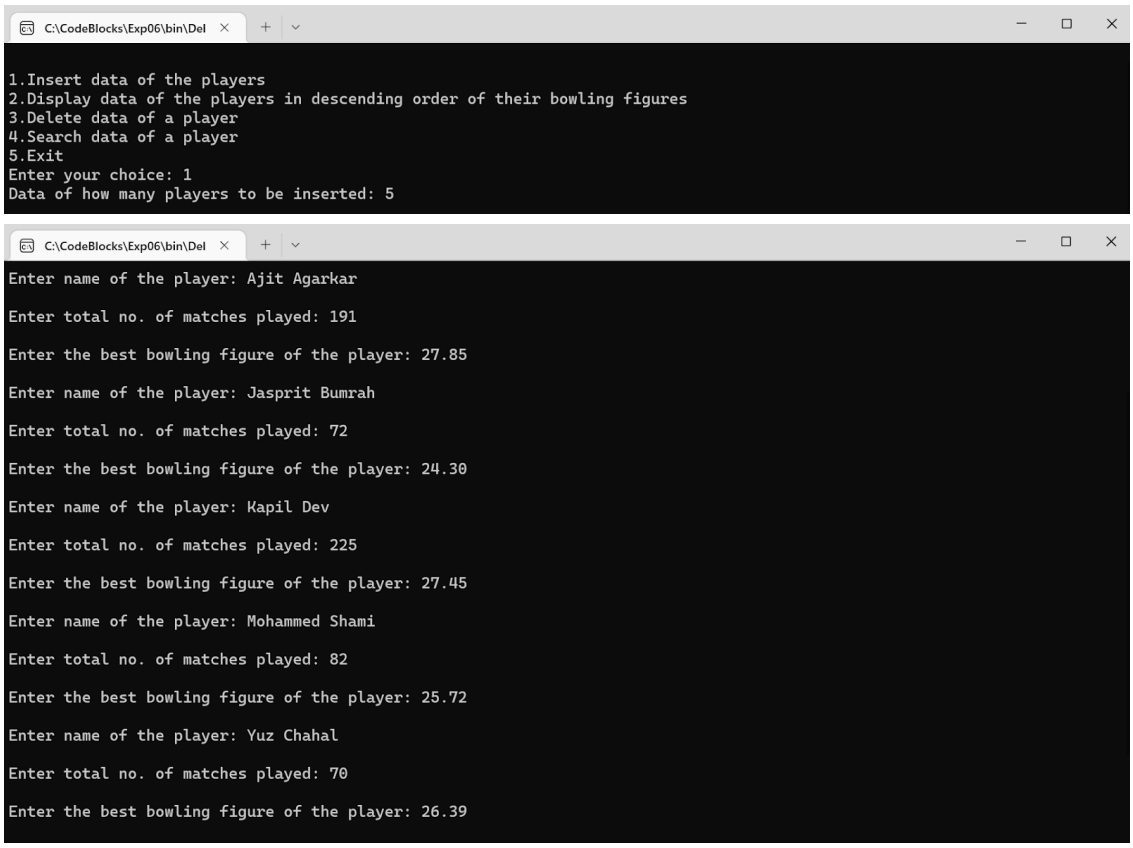
```
#include<stdio.h>
#include<string.h>

struct players
{
   char name[100],surname[100];
   int matches_played;
   float bowling_figure;
}p[100],t;

int n,i,j;

void main()
{
   int choice;
   while(choice!=5)
   {
      printf("\n1.Insert data of the players\n2.Display data of the players in descending order of their bowling figures\n3.Delete data of a player\n4.Search data of a player\n5.Exit");
      printf("\nEnter your choice: ");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1:
            {
               insert();
            }
         break;

         case 2:
            {
               display();
            }
         break;

         case 3:
            {
               delete();
            }
         break;

         case 4:
            {
               search();
```

```
            }
         break;

         case 5:
         break;

         default:
            {
               printf("Enter a valid choice(from 1 to 5 only!)");
            }
         break;

      }
   }
}

void insert()
{
   printf("Data of how many players to be inserted: ");
   scanf("%d",&n);
   for(i=1;i<=n;i++)
   {
      printf("\nEnter name of the player: ");
      scanf("%s %s",p[i].name,p[i].surname);
      printf("\nEnter total no. of matches played: ");
      scanf("%d",&p[i].matches_played);
      printf("\nEnter the best bowling figure of the player: ");
      scanf("%f",&p[i].bowling_figure);
   }
}

void display()
{
   for(i=1;i<=n;i++)
   {
      for(j=1;j<n;j++)
      {
         if(p[j].bowling_figure<p[j+1].bowling_figure)
         {
            t=p[j];
            p[j]=p[j+1];
            p[j+1]=t;
         }
      }
   }
   printf("\nDATABASE OF CRICKET PLAYERS\n");
   printf("\nName\t\t\tTotal Matches Played\t\tBowling Figure\n\n");
```

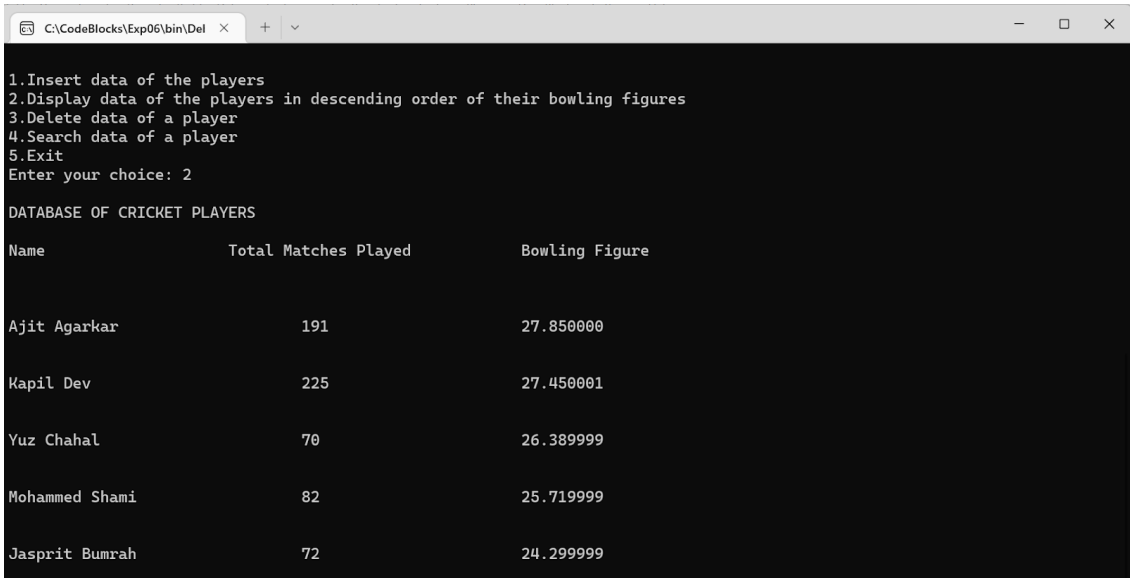
```
   for(i=1;i<=n;i++)
   {
      printf("\n\n%s
%s\t\t\t%d\t\t\t%f\n",p[i].name,p[i].surname,p[i].matches_played,p[i].bowling_figure);
   }
}

void delete()
{
   char delete_name[100],delete_surname[100];
   printf("\nEnter the name of the player whose data is to be deleted: ");
   scanf("%s %s",&delete_name,&delete_surname);
   printf("\nDATABASE OF CRICKET PLAYERS(MODIFIED)\n");
   printf("\nName\t\t\tTotal Matches Played\t\tBowling Figure\n");
   for(i=1;i<=n;i++)
   {
      if(strcmp(delete_name,p[i].name)!=0&&strcmp(delete_surname,p[i].surname)!=0)
      {
         printf("\n\n%s
%s\t\t\t%d\t\t\t%f\n",p[i].name,p[i].surname,p[i].matches_played,p[i].bowling_figure);
      }
   }
}

void search()
{
   char search_name[100],search_surname[100];
   printf("\nEnter the name of the player whose information is to be searched: ");
   scanf("%s %s",&search_name,&search_surname);
   printf("\nINFORMATION ABOUT THE CRICKET PLAYER\n");
   printf("\nName\t\t\tTotal Matches Played\t\tBowling Figure\n");
   for(i=1;i<=n;i++)
   {

if(strcmp(search_name,p[i].name)==0&&strcmp(search_surname,p[i].surname)==0)
      {
         printf("\n%s
%s\t\t\t%d\t\t\t%f\n",p[i].name,p[i].surname,p[i].matches_played,p[i].bowling_figure);
      }
   }
}
```

Output(s):

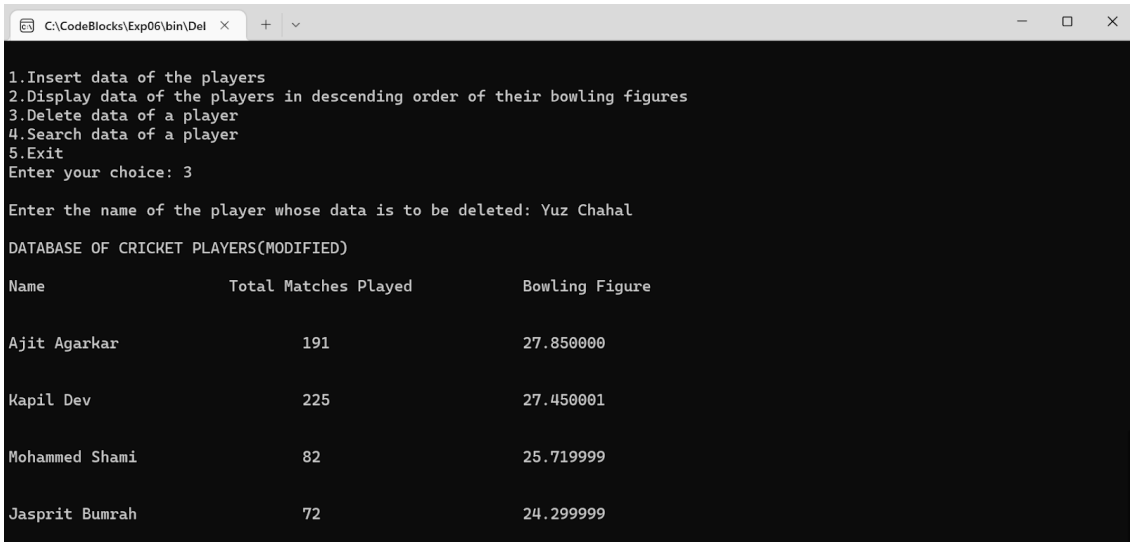
Choice 1:



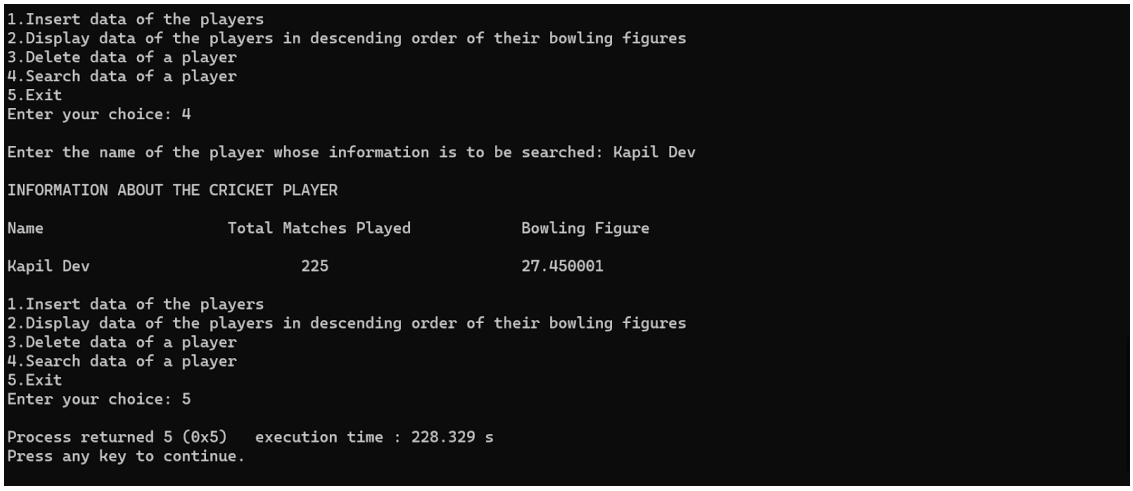
Choice 2:



Choice 3:



Choice 4:



Conclusion:

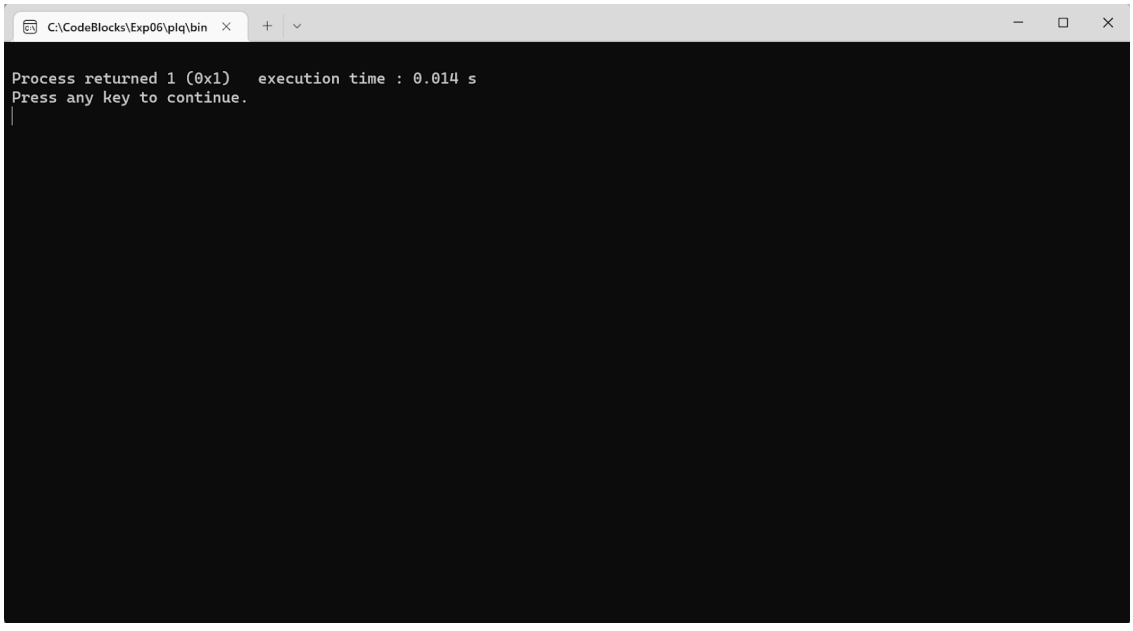
The program could thus call onto various functions in the main function taking input from the user using switch-case about which function to be summoned and running the code in that function. In the first part, the data of various cricket players(their names, total matches played and bowling figures) could be inputted by the user. In the second part, the data was represented in a tabular form and the players were arranged in a descending order of their bowling figures. In the third part, the user could delete the data of any one player; while in the fourth part, the user could search the information about any one player using strcmp (string-compare) command.

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}};
   }
```

Output:



It is an array of structures where the data type is the same for all the members of the structure. As there is no printf function in the code, the output screen is blank.

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: (A) Compiler Error
Field "next" has incomplete type, hence the code will not be compiled and show an error, the compiler pointing to the highlighted line.

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

	STRUCTURE	UNION
Keyword	The keyword **struct** is used to define a structure	The keyword **union** is used to define a union.
Size	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.**
Memory	Each member within a structure is assigned unique storage area of location.	Memory allocated is shared by individual members of union.
Value Altering	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.
Accessing members	Individual member can be accessed at a time.	Only one member can be accessed at a time.
Initialization of Members	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**