Name	Debjit Ghosal
UID no.	2023300065
Experiment No.	8

AIM:	Apply the concepts of structures/union to solve a given problem.
Program 1	
PROBLEM STATEMENT :	A structure “Student” consists of following data: i) student name ii) roll no of student iii) marks Write a programme to read records for n students and perform the following operations using user-defined functions: i) Find student with max marks ii) Sort records according to student name alphabetically iii) Print the sorted records

PSEUDO CODE:	// Define a structure to represent a student
	struct student
	  int rno
	  char name[20]
	  float marks
	
	// Function to read student information from the user
	struct student read_student()
	  struct student s
	  print "Enter the roll no.:"
	  input s.rno
	  flush input buffer
	  print "Enter the name:"
	  input s.name
	  print "Enter the marks:"

```
    input s.marks
    return s

// Function to print student information
void print_student(struct student s)
    print "Roll no.:", s.rno
    print "Name:", s.name
    print "Marks:", s.marks

// Function to sort an array of students based on their names
void sort_students(int n, struct student s[])
    for i from 0 to n-1
        min_idx = i
        for j from i+1 to n-1
            if strcmp(s[j].name, s[min_idx].name) < 0
                min_idx = j
        // Swap the elements to sort the array
        swap s[i] and s[min_idx]

    print "Sorted list is:"
    for i from 0 to n-1
        print "**********************************"
        print_student(s[i])

// Function to find the student with the maximum marks
struct student max_marks(int n, struct student s[])
    max = s[0].marks
    j = 0
    for i from 1 to n-1
        if max < s[i].marks
            max = s[i].marks
            j = i

    print "The maximum marks are obtained by:"
    print "**********************************"
    print_student(s[j])

// Main function
int main()
    n = 0
```

	print "Enter the number of students:"
	input n
	declare an array of students s_arr[n]
	
	// Read student information
	for i from 0 to n-1
	print "**********"
	print "Enter the info of student", i+1, ":"
	s_arr[i] = read_student()
	
	// Print student information
	for i from 0 to n-1
	print "**********"
	print "The info of student", i+1, ":"
	print_student(s_arr[i])
	
	print "**********"
	
	// Sort and print the list of students
	sort_students(n, s_arr)
	
	print "**********"
	
	// Find and print the student with the maximum marks
	max_marks(n, s_arr)
	
	return 0

PROGRAM:	#include<stdio.h>
	#include<string.h>
	struct student
	{
	int rno;
	char name[20];
	float marks;
	};
	struct student read_student()
	{
	struct student s;
	printf("Enter the roll no.:");

```
        scanf("%d", &s.rno);
        getchar();
        printf("Enter the name:");
        scanf("%[^\n]s", s.name);
        printf("Enter the marks:");
        scanf("%f", &s.marks);
        return s;
}
void print_student(struct student s)
{
        printf("Roll no.: %d\n", s.rno);
        printf("Name:%s\n", s.name);
        printf("Marks: %f\n", s.marks);
}
void sort_students (int n,struct student s[])
{
        for(int i=0;i<n;i++)
        {
                int min_idx=i;
                for(int j=i+1;j<n;j++)
                {
                        if(strcmp(s[j].name,s[min_idx].name)<0)
                        min_idx=j;
                }
                struct student temp;
                temp=s[i];
                s[i]=s[min_idx];
                s[min_idx]=temp;
        }
        printf("Sorted list is:\n");
        for (int i=0;i<n;i++)
        {
                printf("**********************************\n");
                print_student(s[i]);
        }
}
struct student max_marks(int n, struct student s[])
{
        int j;
        float max=s[0].marks;
```

```
        for(int i=0;i<n;i++)
                if(max<s[i].marks)
                {
                        max=s[i].marks;
                        j=i;
                }
        printf("the maximum marks are obtained by:\n");
        printf("**********************************\n");
        print_student(s[j]);
}
int main()
{
        int n;
        printf("Enter the number of students:");
        scanf("%d", &n);
        struct student s_arr[n];
        for(int i=0;i<n;i++)
        {
                printf("**********************************\n");
                printf("enter the info of student %d:\n", i+1);
                s_arr[i]=read_student();
        }
        for (int i=0;i<n;i++)
        {
                printf("**********************************\n");
                printf("the info of student %d:\n", i+1);
                print_student(s_arr[i]);
        }
        printf("**********************************\n");
        sort_students (n,s_arr);
        printf("**********************************\n");
        max_marks(n,s_arr);
        return 0;
}
```

RESULT:

```
Enter the number of students:3
********************************
enter the info of student 1:
Enter the roll no.:1
Enter the name:aarayan
Enter the marks:89.5
********************************
enter the info of student 2:
Enter the roll no.:2
Enter the name:zayan
Enter the marks:99.5
********************************
enter the info of student 3:
Enter the roll no.:3
Enter the name:veer
Enter the marks:79.6
********************************
the info of student 1:
Roll no.: 1
Name:aarayan
Marks: 89.500000
********************************
the info of student 2:
Roll no.: 2
Name:zayan
Marks: 99.500000
********************************
the info of student 3:
Roll no.: 3
Name:veer
Marks: 79.599998
********************************
Sorted list is:
********************************
Roll no.: 1
Name:aarayan
Marks: 89.500000
********************************
Roll no.: 3
Name:veer
Marks: 79.599998
********************************
Roll no.: 2
Name:zayan
Marks: 99.500000
********************************
the maximum marks are obtained by:
********************************
Roll no.: 2
Name:zayan
Marks: 99.500000
```

Program 2	
PROBLEM STATEMENT :	A league table consists of a set of N records each representing the performance of a team. A record contains team name, no. of games played, no. of games won, no. of games drawn, no. of games lost, no. of goals scored and no. of points awarded (2 for a win and 1 for a draw). Write a program which inputs a positive integer N, N records of the form above, a positive integer M, the results of M games in the form, team1 goals scored team2 goals scored. Based on the results of these M games, the program should update the records and display the updated records

PSEUDO CODE:	struct team
	  char name[20]
	  int played, won, lost, draw, goals, points
	
	struct match
	  char team1[20], team2[20]
	  int goals1, goals2
	void read_teams(struct team t[], int n)
	  print "Team\tPlayed\tWon\tLost\tDraw\tGoals"
	  for i from 0 to n-1
	  input t[i].name
	  input t[i].played
	  input t[i].won
	  input t[i].lost
	  input t[i].draw
	  input t[i].goals
	  flush input buffer
	  t[i].points = 2 * t[i].won + t[i].draw
	
	void read_matches(struct match g[], int m)
	  print "Team1\tGoals1\tTeam2\tGoals2"
	  for i from 0 to m-1
	  input g[i].team1
	  input g[i].goals1
	  flush input buffer
	  input g[i].team2
	  input g[i].goals2
	  flush input buffer

```
// Function to search for a team in the array
int search(struct team t[], int n, char name[])
    for i from 0 to n-1
        if strcmp(name, t[i].name) == 0
            return i
    return -1

// Function to update team points based on match results
void update_points(int m, int n, struct match g[], struct team t[])
    for i from 0 to m-1
        t1 = search(t, n, g[i].team1)
        t2 = search(t, n, g[i].team2)

        if g[i].goals1 < g[i].goals2
            t[t1].lost++
            t[t2].won++
            t[t2].points += 2
        else if g[i].goals1 > g[i].goals2
            t[t2].lost++
            t[t1].won++
            t[t1].points += 2
        else
            t[t2].draw++
            t[t1].draw++
            t[t2].points += 1
            t[t1].points += 1

        t[t1].played++
        t[t1].goals += g[i].goals1
        t[t2].played++
        t[t2].goals += g[i].goals2

// Function to print the team table
void print_table(struct team t[], int n)
    print "Team\tPlayed\tWon\tLost\tDraw\tGoals\tPoints"
    for i from 0 to n-1
        print t[i].name, t[i].played, t[i].won, t[i].lost, t[i].draw, t[i].goals,
t[i].points
```

	// Main function
	int main()
	  n = 0
	  print "Enter the number of teams:"
	  input n
	  flush input buffer
	  declare an array of teams teams[n]
	
	  // Read team information
	  print "Enter the data of teams:"
	  read_teams(teams, n)
	
	  print "**********"
	  print "The current table is:"
	  print_table(teams, n)
	  print "**********"
	
	  m = 0
	  print "Enter the number of matches:"
	  input m
	  flush input buffer
	  declare an array of matches matches[m]
	
	  // Read match information
	  print "Enter the data of matches:"
	  read_matches(matches, m)
	
	  // Update team points based on match results
	  update_points(m, n, matches, teams)
	
	  print "**********"
	  print "The updated table is:"
	  print_table(teams, n)
	
	  return 0

PROGRAM:	#include<stdio.h> #include<string.h> struct team {

```
        char name[20];
        int played,won,lost,draw,goals,points;
};
struct match
{
        char team1[20], team2[20];
        int goals1, goals2;
};
void read_teams(struct team t[], int n)
{
        printf("\nTeam\tPlayed\tWon\tLost\tDraw\tGoals\n");
        for(int i=0;i<n;i++)
        {
                scanf("%s", t[i].name);
                scanf("%d", &t[i].played);
                scanf("%d", &t[i].won);
                scanf("%d", &t[i].lost);
                scanf("%d", &t[i].draw);
                scanf("%d", &t[i].goals);
                getchar();
                t[i].points=2*t[i].won+t[i].draw;
        }
}
void read_matches(struct match g[], int m)
{
        printf("\nTeam1\tGoals1\tTeam2\tGoals2\n");
        for(int i=0;i<m;i++)
        {
                scanf("%s", g[i].team1);
                scanf("%d", &g[i].goals1);
                getchar();
                scanf("%s", g[i].team2);
                scanf("%d", &g[i].goals2);
                getchar();
        }
}
int search(struct team t[],int n, char name[])
{
        for(int i=0;i<n;i++)
                if(strcmp(name,t[i].name)==0)
```

```
                return i;
        return -1;
}
void update_points(int m, int n, struct match g[], struct team t[])
{
        for (int i=0;i<m;i++)
        {
                int t1=search(t,n,g[i].team1);
                int t2=search(t,n,g[i].team2);
                if(g[i].goals1<g[i].goals2)
                {
                        t[t1].lost++;
                        t[t2].won++;
                        t[t2].points+=2;
                }
                else if(g[i].goals1>g[i].goals2)
                {
                        t[t2].lost++;
                        t[t1].won++;
                        t[t1].points+=2;
                }
                else
                {
                        t[t2].draw++;
                        t[t1].draw++;
                        t[t2].points+=1;
                        t[t1].points+=1;
                }
                t[t1].played++;
                t[t1].goals+=g[i].goals1;
                t[t2].played++;
                t[t2].goals+=g[i].goals2;
        }
}
void print_table(struct team t[],int n)
{
        printf("\nTeam\tPlayed\tWon\tLost\tDraw\tGoals\tPoints\n");
        for(int i=0;i<n;i++)
        {
                printf("\n%s\t%d\t%d\t%d\t%d\t%d\t%d\n",
```

```
t[i].name,t[i].played,t[i].won,t[i].lost,t[i].draw,t[i].goals,t[i].points);
        }

}
int main()
{
        int m, n;
        printf("Enter the number of teams:");
        scanf("%d",&n);
        getchar();
        struct team teams[n];
        printf("Enter the data of teams:");
        read_teams(teams, n);
        printf("**************************************\n");
        printf("The current table is:\n");
        print_table(teams,n);
        printf("**************************************\n");
        printf("Enter the number of matches:");
        scanf("%d",&m);
        getchar();
        struct match matches[m];
        printf("Enter the data of matches:");
        read_matches(matches, m);
        update_points(m,n,matches,teams);
        printf("**************************************\n");
        printf("The updated table is:\n");
        print_table(teams,n);
        return 0;
}
```

RESULT:

```
Enter the number of teams:3
Enter the data of teams:
Team    Played  Won     Lost    Draw    Goals
ABC 5 3 2 0 10
PQR 4 1 2 1 8
XYZ 3 3 0 0 4
************************************
The current table is:

Team    Played  Won     Lost    Draw    Goals   Points

ABC     5       3       2       0       10      6

PQR     4       1       2       1       8       3

XYZ     3       3       0       0       4       6
************************************
Enter the number of matches:3
Enter the data of matches:
Team1   Goals1  Team2   Goals2
ABC 3 PQR 2
PQR 2 XYZ 4
XYZ 2 ABC 3
************************************
The updated table is:

Team    Played  Won     Lost    Draw    Goals   Points

ABC     7       5       2       0       16      10

PQR     6       1       4       1       12      3
```