Name	Lekh Sanatan Nayak
UID no.	202380068
Experiment No.	8

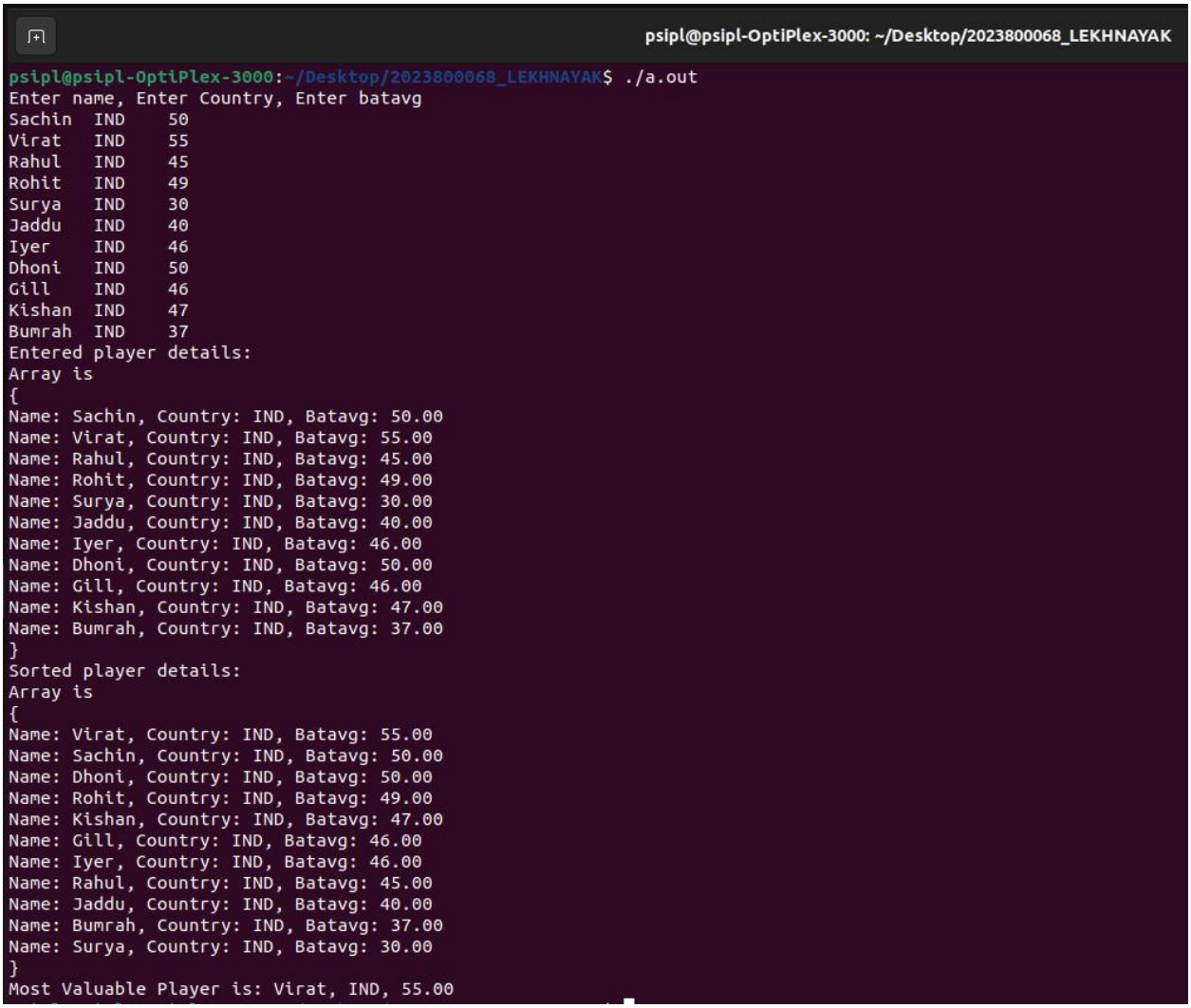
AIM:	**Problem solving using Structures and unions**
Program 1	
PROBLEM STATEMENT :	As a team manager, you want to pick the best batsman from your IPL team for upcoming match. Your system collect data of each player as his name, country and batting average. Then it sorts the list based on batting average and produces the output to you, so you can select best batsman. [Take input of 11 players]
ALGORITHM:	**Define the Player Structure:** 1. Define a structure players with fields name (string), country (string), and batavg (float). **Display Function (display):** 2. Take an array of players and its size as input. 3. Iterate through the array and print the details (name, country, and batting average) for each player. **Swap Function (swap):** 1. Take two pointers to players structures as input. 2. Swap the content of the structures pointed to by the input pointers. **Selection Sort Function (selectionSort):** 1. Take an array of players and its size as input. 2. Implement a selection sort algorithm to sort the array based on batting average in descending order. 3. Use the swap function to swap elements when necessary. **Main Function (main):** 1. Declare an array of players structures (s) with a size of 11. 2. Prompt the user to input details for each player (name, country, and batting average) using a loop and scanf.

	3. Display the entered player details using the display function. 4. Sort the players based on batting average using the selectionSort function. 5. Display the sorted player details using the display function. 6. Print the details of the Most Valuable Player (MVP), which is the player with the highest batting average. **Input/Output:** 1. The user is prompted to enter details for 11 players. 2. The program displays the entered player details. 3. The program then displays the sorted player details based on batting average. 4. Finally, the program prints the details of the Most Valuable Player (MVP).

PROGRAM:	#include<stdio.h>
	#include<string.h>
	
	struct players
	{
	char name[30];
	char country[15];
	float batavg;
	};
	
	void display(struct players a[], int size)
	{
	printf("Array is \n{\n");
	for (int i = 0; i < size; i++)
	{
	printf("Name: %s, Country: %s, Batavg: %.2f\n", a[i].name, a[i].country, a[i].batavg);
	}
	printf("}\n");
	}
	
	void swap(struct players *a, struct players *b)
	{
	struct players temp;
	temp = *a;
	*a = *b;
	*b = temp;
	}

```
void selectionSort(struct players a[], int size)
{
   int min_index;
   for (int i = size - 1; i >= 0; i--)
   {
      min_index = i;
      for (int j = i - 1; j >= 0; j--)
      {
         if (a[j].batavg < a[min_index].batavg)
         {
            min_index = j;
         }
      }
      if (a[i].batavg > a[min_index].batavg)
      {
         swap(&a[i], &a[min_index]);
      }
   }
}

int main()
{
   struct players s[11];

   printf("Enter name, Enter Country, Enter batavg\n");
   for (int i = 0; i < 11; i++)
   {
      scanf("%s", s[i].name);
      scanf("%s", s[i].country);
      scanf("%f", &s[i].batavg);
   }

   printf("Entered player details:\n");
   display(s, 11);

   selectionSort(s, 11);

   printf("Sorted player details:\n");
   display(s, 11);

   printf("Most Valuable Player is: %s, %s, %.2f\n", s[0].name,
s[0].country, s[0].batavg);

   return 0;
}
```

RESULT:



Program 2

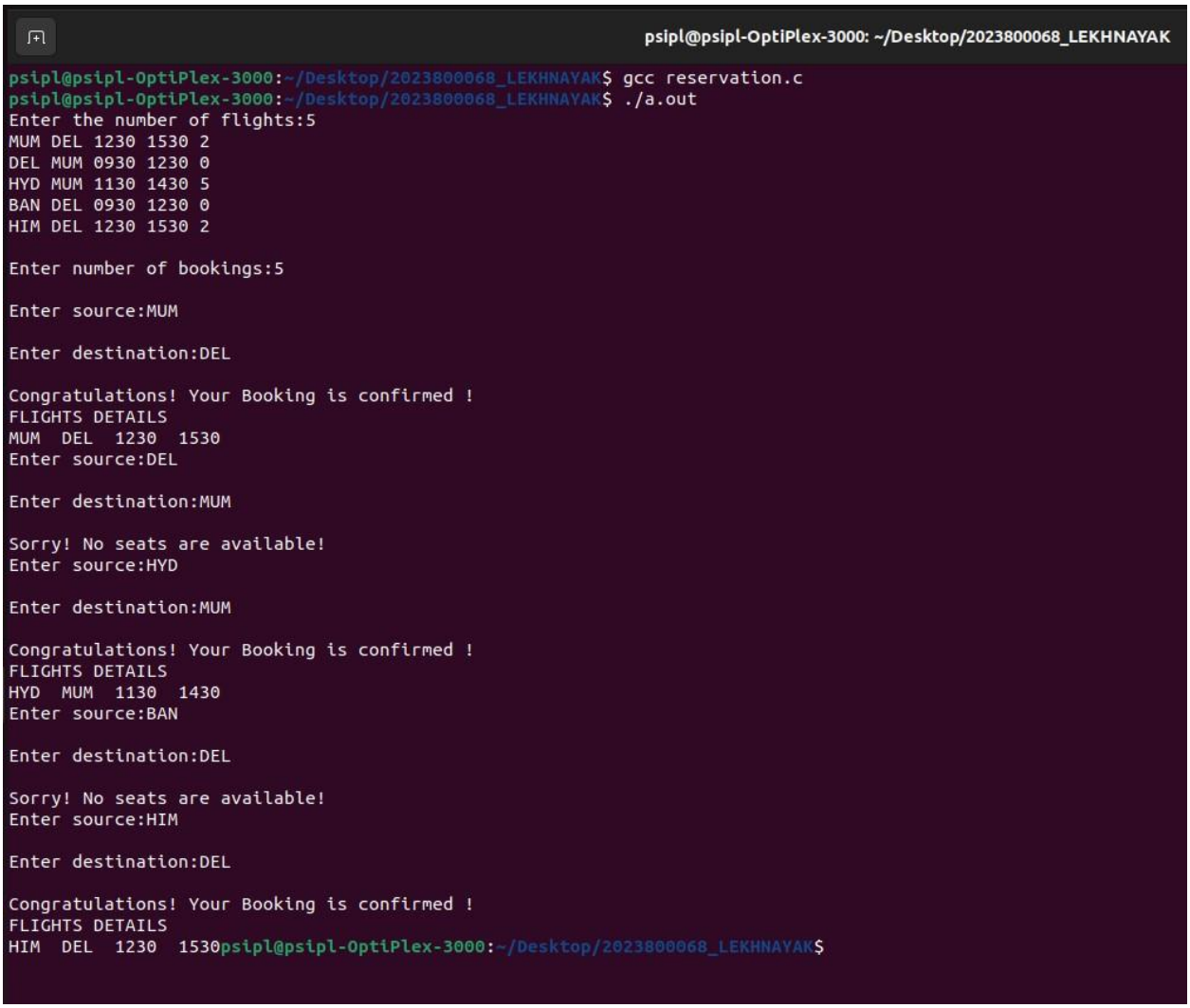
PROBLEM STATEMENT :	An airline reservation system maintains records for possible flights consisting of STARTING POINT 3 character code DESTINATION 3 character code STARTING TIME integer on scale 0001 – 2400 ARRIVAL TIME integer on scale 0001 – 2400 SEATS positive integer in suitable range. Your program is to read 20 such records followed by queries of the form STARTING POINT– DESTINATION, one to a line. For each query find whether there

	is a possible flight with a seat available; if so reduce the number of seats by one and print out the flight details (or an apology).
ALGORITHM:	**Define the flight structure** 1. Create a structure named "flight" and define char variables such as src(source), dest(destination), dt(departure time), at(arrival time). Also define integer variable 'seats' **Declare a function "readflights" with a struct flight f and interger 'n' argument** 1. Take input for an array of 'n' flight structures 2. Use a loop to read source, destination, departure time, arrival time and available seats for each flight **Declare a function "bookflights" with 2 char arguments 'sp' and 'ep' an array of flights (f) and a integer n as arguments** 1. Take source (sp), destination (ep), an array of flights (f), and the number of flights (n) as parameters. 2. Use a loop to iterate through the flights array and check if there is a flight matching the given source and destination. 3. If a match is found and there are available seats, confirm the booking, decrease the available seats, and set the flag to 1. 4. If no matching flight or no available seats, set flag to 0. 5. Print a confirmation or rejection message based on the value of the flag **Declare a function "main"** 1. Take input for the number of flights (n). 2. Declare an array of n flight structures named flights. 3. Call the "readflights" function to input details for each flight. 4. Take input for the number of bookings (m). 5. Use a loop to input source and destination for each booking and call the "bookflight" function. 6. The program prompts the user to enter the number of flights, details for each flight, and the number of bookings. **7.** For each booking, the program asks for the source and destination, checks for available seats on matching flights, and prints a confirmation or rejection message
PROGRAM:	#include<stdio.h> #include<string.h> struct flight{

```
        char src[4],dest[4],dt[5],at[5];
        int seats;
};

void readflights(struct flight f[], int n){
        for (int i=0;i<n;i++){
                scanf("%s",f[i].src);
                scanf("%s",f[i].dest);
                scanf("%s",f[i].dt);
                scanf("%s",f[i].at);
                scanf("%d",&f[i].seats);
        }
}

void bookflight(char sp[],char ep[], struct flight f[], int n){
        int flag = 0;
        for(int i=0;i<n;i++){
        if(strcmp(sp, f[i].src)==0 && strcmp(ep, f[i].dest)==0){
                if(f[i].seats){
                        flag=1;
                        printf("\nCongratulations! Your Booking is
confirmed !");
                        printf("\nFLIGHTS DETAILS \n%s  %s  %s  %s",
sp, ep, f[i].dt, f[i].at);
                        f[i].seats--;
                }
        }
  }
  if(flag==0){
        printf("\nSorry! No seats are available!");
  }

}

int main(){
        int n,m;
        printf("Enter the number of flights:");
        scanf("%d",&n);
        struct flight flights[n];
```

```
        readflights(flights,n);

        printf("\nEnter number of bookings:");
        scanf("%d",&m);
        char sp[4], ep[4];

        for(int i=0;i<m;i++){
                printf("\nEnter source:");
                scanf("%s",sp);
                printf("\nEnter destination:");
                scanf("%s",ep);
                bookflight(sp,ep,flights,n);
        }
        return 0;
}
```

RESULT:



CONCLUSION:	In this experiment I learnt how to use structures and unions to solve problem statements