

<b>Name</b>	Jhaveri Varun Nimitt
<b>UID no.</b>	2023800042
<b>Experiment No.</b>	8

<b>AIM:</b>	Using structures and unions to solve a given problem.
<b>Program 1</b>	
<b>PROBLEM STATEMENT :</b>	As a team manager, you want to pick the best batsman from your IPL team for an upcoming match. Your system collects 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 the best batsman. [Take input of 11 players]
<b>ALGORITHM:</b>	<p><b>START</b></p> <p>1.1 Define a structure named player with fields:</p> <ul style="list-style-type: none"> <li>- name (string)</li> <li>- country (string)</li> <li>- battingavg (double)</li> </ul> <p>2.1 Function selectionSort(arr: array of player, n: integer)</p> <p>2.1.1 For i from 0 to n-1</p> <p>2.1.1.1 Set max_idx to i</p> <p>2.1.1.2 For j from i+1 to n</p> <p>2.1.1.2.1 If arr[j].battingavg &gt; arr[max_idx].battingavg</p> <p style="padding-left: 40px;">Set max_idx to j</p> <p>2.1.1.3 Swap arr[max_idx] and arr[i]</p> <p>3.1 Declare an array of player structures named allmyplayers with size 11</p> <p>3.2 For i from 0 to 11; i &lt; 11; i++</p> <p>3.2.1 Prompt the user to input player data:</p> <ul style="list-style-type: none"> <li>- Enter name</li> <li>- Enter country</li> <li>- Enter batting average</li> <li>- Store input in allmyplayers[i].name, allmyplayers[i].country, allmyplayers[i].battingavg</li> </ul> <p>3.3 Call selectionSort(allmyplayers, 11) to sort the players based on batting averages</p> <p>4.1 For i from 0 to 11; i &lt; 11; i++</p> <p>4.1.1 Print allmyplayers[i].name, allmyplayers[i].country, allmyplayers[i].battingavg</p> <p>5.1 Print "The best batsman from your IPL team for upcoming match is: "</p> <p>5.2 Print allmyplayers[0].name</p> <p><b>END</b></p>

**PROGRAM:**

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

struct player {
    char name[20];
    char country[20];
    double battingavg;
};

void selectionSort(struct player arr[], int n)
{
    int i, j, max_idx;
    struct player temp;
    for (i = 0; i < n-1; i++)
    {
        max_idx = i;
        for (j = i+1; j < n; j++)
        {
            if (arr[j].battingavg > arr[max_idx].battingavg)
            {
                max_idx = j;
            }
        }
        temp = arr[max_idx];
        arr[max_idx] = arr[i];
        arr[i] = temp;
    }
}

int main()
{
    struct player allmyplayers[11];
    printf("Enter name, country and batting average for all 11 players
(separated by a space): \n");
    for (int i = 0; i < 11; i++)
    {
        scanf("%s", allmyplayers[i].name);
        scanf("%s", allmyplayers[i].country);
        scanf("%lf", &allmyplayers[i].battingavg);
    }
    selectionSort(allmyplayers, 11);
    printf("\nSorted players by batting average:\n");
    for (int i = 0; i < 11; i++)
    {
        printf("%s (%s): %.2lf", allmyplayers[i].name, allmyplayers[i].country,
allmyplayers[i].battingavg);
        printf("\n");
    }

    printf("The best batsman from your IPL team for upcoming match is: %s",
```

```
allmyplayers[0].name);  
    return 0;  
}  
/*  
Harry AUS 34.56  
Hitesh IND 78.90  
Kanye WI 12.34  
Hardik IND 56.78  
Virat IND 90.12  
Babar PAK 45.67  
Sachin IND 89.01  
Kautubh JAP 23.45  
Kshitij MLM 67.89  
Marshall NZ 45.67  
Sindhu IND 78.90  
*/
```

## RESULT:

```
cyclops@cyclops: ~/Desktop/PSIPL Semester 1/Experiment 8  
cyclops@cyclops: ~/Desktop/PSIPL Semester 1/Experiment 8$ gcc iplsorter.c  
cyclops@cyclops: ~/Desktop/PSIPL Semester 1/Experiment 8$ ./a.out  
Enter name, country and batting average for all 11 players (separated by a space):  
Harry AUS 34.56  
Hitesh IND 78.90  
Kanye WI 12.34  
Hardik IND 56.78  
Virat IND 90.12  
Babar PAK 45.67  
Sachin IND 89.01  
Kautubh JAP 23.45  
Kshitij MLM 67.89  
Marshall NZ 45.67  
Sindhu IND 78.90  
  
Sorted players by batting average:  
Virat (IND): 90.12  
Sachin (IND): 89.01  
Hitesh (IND): 78.90  
Sindhu (IND): 78.90  
Kshitij (MLM): 67.89  
Hardik (IND): 56.78  
Marshall (NZ): 45.67  
Babar (PAK): 45.67  
Harry (AUS): 34.56  
Kautubh (JAP): 23.45  
Kanye (WI): 12.34  
The best batsman from your IPL team for upcoming match is: Viratcyclops@cyclops: ~/Desktop/PSIPL Semester 1/Experiment 8$
```

Program 2	
<b>PROBLEM STATEMENT :</b>	<p>An airline reservation system maintains records for possible flights consisting of:</p> <p>STARTING POINT 3 character code</p> <p>DESTINATION 3 character code</p> <p>STARTING TIME integer on scale 0001 – 2400</p> <p>ARRIVAL TIME integer on scale 0001 – 2400</p> <p>SEATS positive integer in suitable range.</p> <p>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).</p>
<b>ALGORITHM:</b>	<p><b>START</b></p> <p>1.1 Define a structure named Flight with fields:</p> <ul style="list-style-type: none"> <li>- og (string): Departure city</li> <li>- dest (string): Arrival city</li> <li>- <b>start_time</b> (integer): Departure time</li> <li>- arrival_time (integer): Arrival time</li> <li>- seats (integer): Number of seats in the flight</li> </ul> <p>1.2 Define color constants for formatting output:</p> <ul style="list-style-type: none"> <li>- COLOR_BOLD_SLOW_BLINKING_RED</li> <li>- COLOR_BOLD_SLOW_BLINKING_BLUE</li> <li>- COLOR_OFF</li> <li>- COLOR_BOLD</li> </ul> <p>2.1 Function get_flight_details(flights: array of Flight, n: integer)</p> <p>2.1.1 Display prompt: "Enter details for all 20 flights in the format of Departure city, Arrival city, Departure time, Arrival time, Number of seats in flight:"</p> <p>2.1.2 For i from 0 to n; i++</p> <p>2.1.2.1 Read input for flights[i].og, flights[i].dest, flights[i].start_time, flights[i].arrival_time, flights[i].seats</p> <p>2.1.2.2 If flights[i].start_time == 2400 or flights[i].arrival_time == 2400, display error and terminate program</p> <p>3.1 Function book_flight(flights: array of Flight, n: integer)</p> <p>3.1.1 While true</p> <p>3.1.1.1 Display prompt: "Enter your query in the format of   Departure City Arrival City  :"</p> <p>3.1.1.2 Read input for wherego and whereland</p> <p>3.1.1.3 For j from 0 to n ; j++</p> <p>3.1.1.3.1 If strcmp(flights[j].og == wherego and flights[j].dest == whereland and flights[j].seats &gt; 0) ==0</p> <p>3.1.1.3.1.1 Decrement flights[j].seats</p> <p>3.1.1.3.1.2 Display booking confirmation with formatted information</p> <p>3.1.1.3.1.3 Break from the loop</p> <p>3.1.1.4 If j == n, display "No flights available from wherego to whereland"</p> <p>3.1.1.5 Display prompt: "Press C to end, otherwise press D to start another query:"</p> <p>3.1.1.6 Read input for exitcheck</p> <p>3.1.1.7 If exitcheck == 'C' or 'c', break from the loop</p> <p>4.1 Function main()</p>

	<p>4.1.1 Declare an array of Flight structures named flights with a size of 20</p> <p>4.1.2 Set n to 20</p> <p>4.1.3 Call get_flight_details(flights, n) to input flight details</p> <p>4.1.4 Call book_flight(flights, n) to handle flight bookings</p> <p><b>END</b></p>
<p><b>PROGRAM:</b></p>	<pre> #include &lt;stdio.h&gt; #include &lt;string.h&gt; #include &lt;stdlib.h&gt; #define COLOR_BOLD_SLOW_BLINKING_RED "\e[1;5;31m" #define COLOR_BOLD_SLOW_BLINKING_BLUE "\e[1;5;34m" #define COLOR_OFF "\e[m" #define COLOR_BOLD "\e[1m"  struct Flight {     char og[20];     char dest[20];     int start_time;     int arrival_time;     int seats; };  void get_flight_details(struct Flight flights[], int n) {     int i;     printf("Enter details for all 20 flights in the format of Departure city, Arrival city, Departure time, Arrival time, Number of seats in flight: \n");     for (i = 0; i &lt; n; i++)     {         scanf("%3s %3s %d %d %d", flights[i].og, flights[i].dest, &amp;flights[i].start_time, &amp;flights[i].arrival_time, &amp;flights[i].seats);         if(flights[i].start_time == 2400    flights[i].arrival_time==2400)         {             printf("Error: Time cannot be above 2400 or 2400, terminating program.");             exit(0);         }     } }  void book_flight(struct Flight flights[], int n) {     int j;     char wherego[20], whereland[20];     char exitcheck;      while (1)     {         printf("\nEnter your query in the format of   Departure City Arrival City  :");     } } </pre>

```

        scanf("%3s %3s", wherego, whereand);
        for (j = 0; j < n; j++)
        {
            if (strcmp(flights[j].og, wherego) == 0 && strcmp(flights[j].dest,
whereand) == 0 && flights[j].seats > 0)
            {
                flights[j].seats--;
                printf(COLOR_BOLD"Booking Confirmed:\n\n"COLOR_OFF);
                printf("Your flight departs from");
                printf(COLOR_BOLD_SLOW_BLINKING_RED" %s ",flights[j].og );
                printf(COLOR_OFF "at");
                printf(COLOR_BOLD_SLOW_BLINKING_BLUE" %d
",flights[j].start_time);
                printf(COLOR_OFF"and arrives at");
                printf(COLOR_BOLD_SLOW_BLINKING_RED" %s ",flights[j].dest);
                printf(COLOR_OFF "at");
                printf(COLOR_BOLD_SLOW_BLINKING_BLUE"
%d.",flights[j].arrival_time);
                printf(COLOR_OFF "\n\nThe amount of seats left in the same
flight is: %d\n", flights[j].seats);
                break;
            }
        }
        if (j == n)
        {
            printf("\nNo flights available from %s to %s\n", wherego,
whereand);
        }

        printf("\nPress C to end, otherwise press D to start another query: ");
        scanf(" %c", &exitcheck);
        if (exitcheck == 'C' || exitcheck == 'c')
        {
            break;
        }
        else
        {
            continue;
        }
    }
}

int main()
{
    struct Flight flights[20];
    int n = 20;

    get_flight_details(flights, n);
    book_flight(flights, n);

    return 0;
}

```

```
Applications  Places  Asaka | SINce Memories ... | Hikarito kageno laplace  Fri Nov 10 22:33:53
cyclops@cyclops: ~/Desktop/PSIPL Semester 1/Experiment 8
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 8$ ./a.out
Enter details for all 20 flights in the format of Departure city, Arrival city, Departure time, Arrival time, Number of seats in flight:
LAX JFK 1330 2100 1
ORD ATL 1400 2000 2
SFO MIA 1500 2300 3
DFW SEA 1600 2200 4
LAS BOS 1700 0300 5
MSP SLC 1530 2200 6
IAH DCA 1430 2100 7
DEN PHL 1530 2300 8
PHX DTW 1400 2200 9
MCO IAD 1500 2300 10
EWR LGA 1430 1800 11
LGA ORD 1500 1800 12
SEA MCO 1400 2200 13
ATL DFW 1530 2000 1
JFK LAS 1600 2300 2
MIA MSP 1630 2300 3
BOS PHX 1400 2200 4
SLC IAH 1430 2100 5
PHL DEN 1500 2300 6
DCA SFO 1330 2100 7

Enter your query in the format of | Departure City Arrival City |:MIA MSP
Booking Confirmed:

Your flight departs from MIA at 1630 and arrives at MSP at 2300.

The amount of seats left in the same flight is: 2

Press C to end, otherwise press D to start another query: d

Enter your query in the format of | Departure City Arrival City |:SLC IAH
Booking Confirmed:

Your flight departs from SLC at 1430 and arrives at IAH at 2100.

The amount of seats left in the same flight is: 4

Press C to end, otherwise press D to start another query: d

Enter your query in the format of | Departure City Arrival City |:BOS PHX
Booking Confirmed:

Your flight departs from BOS at 1400 and arrives at PHX at 2200.

The amount of seats left in the same flight is: 3

Press C to end, otherwise press D to start another query: c
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 8$
```

RESULT:

Better resolution (show the blinking text) to showcase the cool stuff I found at : [bash commands](#)

CONCLUSION:	I have understood the difference between structures and unions and learned how to use them to solve a problem efficiently.
-------------	--