

Name: SUNDEEP A	SRN:PES1UG20CS445	Section: O
	Date:17/6/2021	Week Number:7

1	<p>Define a structure called cricket that will describe the following information:</p> <p style="padding-left: 40px;">player name</p> <p style="padding-left: 40px;">team name</p> <p style="padding-left: 40px;">batting average</p> <p>Using cricket, declare an array player with 5 elements and write a program to read the information about all the 5 players and print a team-wise list containing names of player with their batting average. Write functions for the following:</p> <p>i) Read the information of all the 5 players</p> <p>ii) Sorting the players</p> <p>iii) Displaying team-wise list containing names of player with their batting average</p> <p>Input:</p> <p>Enter data of 5 players</p> <p>Enter PName TName BAvg for player-1 = sachin</p> <p>India</p> <p>98</p> <p>Enter PName TName BAvg for player-2 = Rahul</p> <p>India</p> <p>45</p> <p>Enter PName TName BAvg for player-3 = Jonty</p> <p>Australia</p>
---	--

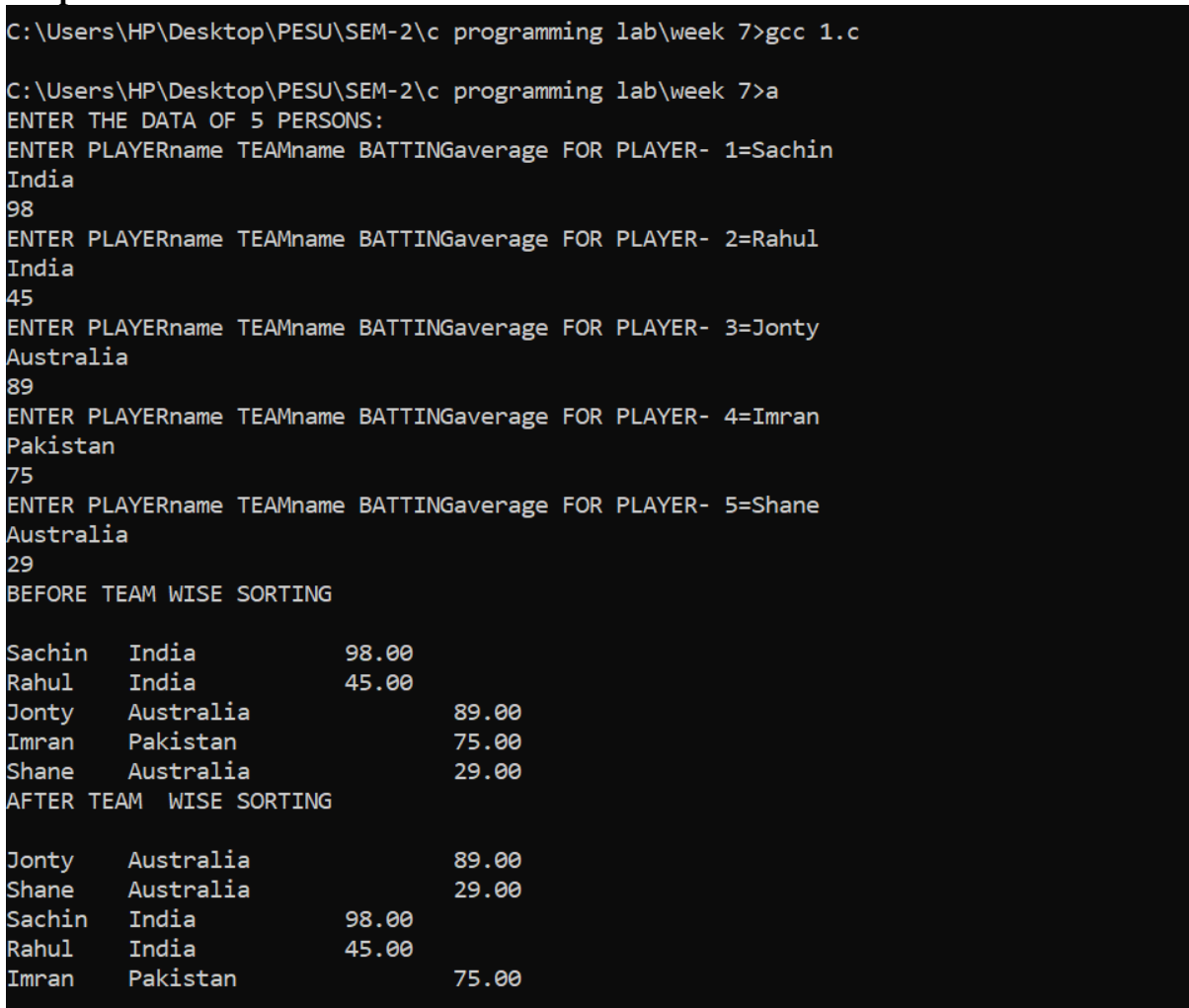
	<p>89</p> <p>Enter PName TName BAvG for player-4 = Imran</p> <p>pakistan</p> <p>75</p> <p>Enter PName TName BAvG for player-5 = Shen</p> <p>Australia</p> <p>29</p> <p>Output:</p> <p>After teamwise sorting... Player list is</p> <table><tr><td>Jonty</td><td>Australia</td><td>89.00</td></tr><tr><td>Shen</td><td>Australia</td><td>29.00</td></tr><tr><td>sachin</td><td>India</td><td>98.00</td></tr><tr><td>Rahul</td><td>India</td><td>45.00</td></tr><tr><td>Imran</td><td>pakistan</td><td>75.00</td></tr></table>	Jonty	Australia	89.00	Shen	Australia	29.00	sachin	India	98.00	Rahul	India	45.00	Imran	pakistan	75.00
Jonty	Australia	89.00														
Shen	Australia	29.00														
sachin	India	98.00														
Rahul	India	45.00														
Imran	pakistan	75.00														
	<p>Program:</p> <pre>#include<stdio.h> #include<conio.h> #include<string.h> //structure for holding players details struct cricket{ char pname[20]; char tname[20]; float bavg; }; typedef struct cricket crick; void readp(crick *p); void teamsort(crick *p);</pre>															

```
void displayp(crick *p);
int main()
{
    crick player[5];
    readp(player);
    printf("BEFORE TEAM WISE SORTING\n\n");
    displayp(player);
    teamsort(player);
    printf("AFTER TEAM WISE SORTING\n\n");
    displayp(player);
    return 0;
}

//Input function
void readp(crick *p)
{
    printf("ENTER THE DATA OF 5 PERSONS:\n");
    for(int j=0;j<5;j++)
    {
        printf("ENTER PLAYERname TEAMname BATTINGaverage FOR\n\n");
        printf("PLAYER- %d=",j+1);
        scanf("%s",&p[j].pname);
        scanf("%s",&p[j].tname);
        scanf("%f",&p[j].bavg);
    }
}

//Sorting function based on teams name
void teamsort(crick *p)
{
    crick temp;
    for(int j=0;j<5;j++)
    {
        for(int k=0;k<4-j;k++)
        {
            if(strcmp(p[k].tname,p[k+1].tname)>0)
            {
                temp=p[k];
                p[k]=p[k+1];
                p[k+1]=temp;
            }
        }
    }
}

//Display function
```

	<pre>void displayp(crick *p) { for(int j=0;j<5;j++) { printf("%s\t %s\t\t %.2f\n",p[j].pname,p[j].tname,p[j].bavg); } }</pre>
	<p>Output Screenshot:</p>  <pre>C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\week 7>gcc 1.c C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\week 7>a ENTER THE DATA OF 5 PERSONS: ENTER PLAYERname TEAMname BATTINGaverage FOR PLAYER- 1=Sachin India 98 ENTER PLAYERname TEAMname BATTINGaverage FOR PLAYER- 2=Rahul India 45 ENTER PLAYERname TEAMname BATTINGaverage FOR PLAYER- 3=Jonty Australia 89 ENTER PLAYERname TEAMname BATTINGaverage FOR PLAYER- 4=Imran Pakistan 75 ENTER PLAYERname TEAMname BATTINGaverage FOR PLAYER- 5=Shane Australia 29 BEFORE TEAM WISE SORTING Sachin India 98.00 Rahul India 45.00 Jonty Australia 89.00 Imran Pakistan 75.00 Shane Australia 29.00 AFTER TEAM WISE SORTING Jonty Australia 89.00 Shane Australia 29.00 Sachin India 98.00 Rahul India 45.00 Imran Pakistan 75.00</pre>
2	<p>Implement Priority Queue using an Unordered Linked list.</p> <p>Write functions for the following</p>

1)Initialization

2)Enqueue

3)Dequeue

4)Display

Output:

enter ua choice

1.insert 2.delete 3.display 4 exit

1

enter the detail and priority

10

1

enter ua choice

1.insert 2.delete 3.display 4 exit

1

enter the detail and priority

20

2

enter ua choice

1.insert 2.delete 3.display 4 exit

1

enter the detail and priority

30
3
enter ua choice
1.insert 2.delete 3.display 4 exit
3
30 3
20 2
10 1
enter ua choice
1.insert 2.delete 3.display 4 exit
1
enter the detail and priority
40
0
enter ua choice
1.insert 2.delete 3.display 4 exit
3
40 0
30 3
20 2
10 1

enter ua choice

1.insert 2.delete 3.display 4 exit

2

deleted node detail is 30 with priority 3

enter ua choice

1.insert 2.delete 3.display 4 exit

2

deleted node detail is 20 with priority 2

enter ua choice

1.insert 2.delete 3.display 4 exit

2

deleted node detail is 10 with priority 1

enter ua choice

1.insert 2.delete 3.display 4 exit

2

deleted node detail is 40 with priority 0

enter ua choice

1.insert 2.delete 3.display 4 exit

2

no elements to delete

enter ua choice

	<p>1.insert 2.delete 3.display 4 exit</p> <p>4</p>
	<p>Program:</p> <pre> #include<stdio.h> #include<stdlib.h> #include<string.h> //structure used to store the details and priority typedef struct component { char details[20]; int priority; }compo; //used for linking struct node { compo c; struct node *link; }; typedef struct node node_t; struct priority_queue { node_t *head; }; typedef struct priority_queue prio_Q; void init(prio_Q* b); void EQ(prio_Q* b,compo* com); void DQ(prio_Q* b); void disp(const prio_Q* b); int main() { prio_Q q; init(&q); compo c; printf("Enter your choice 1.insert 2.delete 3.disp 4.exit\n"); int choice; scanf("%d",&choice); do{ switch(choice){ case 1: printf("Enter data and priority : "); scanf("%s %d",c.details,&c.priority); </pre>


```

        EQ(&q,&c);
        break;
    case 2: DQ(&q);
        break;
    case 3: disp(&q);
        break;
    default: return 0;
}
printf("Enter your choice 1.insert 2.delete 3.disp 4.exit\n");
scanf("%d",&choice);
}while(choice!=4);
return 0;
}

//initialization
void init(prio_Q* b)
{
    b->head=NULL;
}

//enqueue
void EQ(prio_Q* b,compo* com)
{
    node_t* temp=(node_t*)malloc(sizeof(node_t));
    strcpy(temp->c.details,com->details);
    temp->c.priority=com->priority;
    temp->link=b->head;
    b->head=temp;
}

//Dequeue
void DQ(prio_Q* b)
{
    if(b->head==NULL)
        printf("NO elements to delete\n");
    else
    {
        node_t* present=b->head;
        node_t* prev=NULL;
        int max=0;
        node_t* prev_max=NULL;
        while(present!=NULL)
        {
            if(present->c.priority>=max)
            {
                max=present->c.priority;
            }
        }
    }
}

```

	<pre> prev_max=prev; } prev=present; present=present->link; } compo co; if(prev_max!=NULL) { node_t* temp=prev_max->link; prev_max->link=temp->link; strcpy(co.details,temp->c.details); co.priority=temp->c.priority; free(temp); } else { node_t* temp=b->head; b->head=b->head->link; strcpy(co.details,temp->c.details); co.priority=temp->c.priority; free(temp); } printf("Deleted node details is %s with %d priority\n",co.details,co.priority); } } //Display void disp(const prio_Q* b) { node_t* p=b->head; if(p==NULL) printf("No elements in queue\n"); else { while(p!=NULL) { printf("%s %d\n",p->c.details,p->c.priority); p=p->link; } } } </pre>
	Output Screenshot:

```
C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\week 7>gcc 2.c
```

```
C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\week 7>a
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
1
```

```
Enter data and priority : 10
```

```
1
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
1
```

```
Enter data and priority : 20
```

```
2
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
1
```

```
Enter data and priority : 30
```

```
3
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
3
```

```
30 3
```

```
20 2
```

```
10 1
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
1
```

```
Enter data and priority : 40
```

```
0
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
3
```

```
40 0
```

```
30 3
```

```
20 2
```

```
10 1
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
2
```

```
Deleted node details is 30 with 3 priority
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
2
```

```
Deleted node details is 20 with 2 priority
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
2
```

```
Deleted node details is 10 with 1 priority
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
2
```

```
Deleted node details is 40 with 0 priority
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
2
```

```
NO elements to delete
```

```
Enter your choice 1.insert 2.delete 3.disp 4.exit
```

```
4
```

