Week 7: Programs on Array Of Structures And Linked List

2021

Name:AYUSH SINGH	SRN:PES2UG20S081	Section:B
	Date:27/06/2002	Week Number:7

1	Define a structure called cricket that will describe the following information: player name team name batting average 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: i) Read the information of all the 5 players ii)Sorting the players iii)Displaying team-wise list containing names of player with their batting average **Input:** Enter data of 5 players Enter PName TName BAvg for player-1 = sachin India 98 Enter PName TName BAvg for player-2 = Rahul India 45 Enter PName TName BAvg for player-3 = Jonty Australia 89 Enter PName TName BAvg for player-4 = Imran pakistan 75 Enter PName TName BAvg for player-5 = Shen

Australia

29

Output:

After teamwise sorting... Player list is

Jonty	Australia	89.00
Shen	Australia	29.00
sachin	India	98.00
Rahul	India	45.00
Imran	pakistan	75.00

Program:

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

struct cricket{
   char pname[20];char tname[20];float bavg;
};

void main(){
   struct cricket s[5],t;
   int n=5;
   float avg;

   printf("Enter Player Data\n",n);
   for (int i=0;i<n;i++){
      printf("Enter Name, Team and Batting Average for Player %d: ",i+1);
      scanf("%s %s %f",s[i].pname, s[i].tname, &avg);
      s[i].bavg = avg;
   }

   for(int i=1;i<=n-1;i++){
      for(int j=1;j<=n-i;j++){
         if(strcmp(s[j-1].tname,s[j].tname)>0){
            t=s[j-1];
            s[j-1]= s[j];
            s[j] = t;
         }

      }
   }
```

```
    printf("\nPlayer List After Sorting According to Team");
    for(int i=0;i<n;i++){
        printf("\n%-20s %-20s %.2f",s[i].pname,s[i].tname,s[i].bavg);
    }
}
```

Output Screenshot:

```
PS D:\Important Files\AYUSHSINGHPES2UG20CS081\WEEK7> cd "d:\Important Files\AYUS
HSINGHPES2UG20CS081\WEEK7\" ; if ($?) { gcc bruh1.c -o bruh1 } ; if ($?) { .\bru
h1 }
Enter Player Data
Enter Name, Team and Batting Average for Player 1: n1 t1 23
Enter Name, Team and Batting Average for Player 2: n2 t1 34
Enter Name, Team and Batting Average for Player 3: n3 t5 21
Enter Name, Team and Batting Average for Player 4: n4 t6 11
Enter Name, Team and Batting Average for Player 5: n6 t7 99

Player List After Sorting According to Team
n1                   t1                   23.00
n2                   t1                   34.00
n3                   t5                   21.00
n4                   t6                   11.00
n6                   t7                   99.00
PS D:\Important Files\AYUSHSINGHPES2UG20CS081\WEEK7>
```

2 **Implement Priority Queue using an Unordered Linked list.**

Write functions for the following

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
1.insert 2.delete 3.display 4 exit
4
```

Program:

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


typedef struct node{
   int info,priority;
   struct node* link;
}NODE;

void insert(int,int);
void del();
void display();
NODE *front = NULL;


void main(){


   int choice,item,priority;

   do{
      printf("\n****** Queue Operations ******\n");
      printf("1. Insert\n");
      printf("2. Delete\n");
      printf("3. Display\n");
      printf("4. Quit\n");
      printf("Enter Choice: ");
      scanf("%d",&choice);

      switch (choice){
         case 1:
            printf("Value & Priority: ");
            scanf("%d %d",&item,&priority);
            insert(item,priority);
            break;
```

```
            case 2:
               del();
               break;
            case 3:
               display();
               break;
            case 4:
               break;
            default:
               printf("Invalid Choice\n");
         }
      }
      while(choice!=4);
   }


   void insert(int value,int rank){
      NODE *tmp, *x;

      tmp = (NODE*)malloc(sizeof(NODE));
      tmp->info = value;
      tmp->priority = rank;

      if (front == NULL || rank < front->priority){
         tmp->link = front;
         front  = tmp;
      }
      else{
         x = front;
         while(x->link != NULL && x->link->priority <= rank)
            x = x->link;
         tmp->link = x->link;
         x->link = tmp;
      }
   }

   void del(){
      NODE *tmp;
      if (front==NULL)
         printf("Queue Underflow\n");
      else{
         tmp = front;
         printf("Deleted Value %d with priority %d..\n",tmp->info,tmp->priority);
         front = front->link;
         free(tmp);
      }
   }
```

```
void display(){
   NODE *ptr;
   ptr = front;

   if (front == NULL){
      printf("Queue is Empty\n");
      return; //a
   }
   printf("[");
   while(ptr->link !=NULL){
      printf("%d(%d)-->",ptr->info,ptr->priority);
      ptr = ptr->link;
   }
   printf("%d(%d)]\n",ptr->info,ptr->priority);
}
```

Output Screenshot:

```
****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 2
Queue Underflow

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 3
Queue is Empty

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 1
Value & Priority: 5 0

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 1
Value & Priority: 4 6

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 1
Value & Priority: 1 9
```

```
****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 3
[5(0)-->4(6)-->1(9)]

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 2
Deleted Value 5 with priority 0..

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 2
Deleted Value 4 with priority 6..

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 3
[1(9)]

****** Queue Operations ******
1. Insert
2. Delete
3. Display
4. Quit
Enter Choice: 4
```