

**NAME – RAJDEEP JAISWAL**

**DATE – 19 NOV 2021**

**BRANCH – BTECH CSE**

**SEC = 608 - A**

**UID -20BCS2761**

**Subject – DS Lab**

### **AIM –**

1. Program to implement Queue using linked list.

### **CODE IN TEXT FORM –**

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *front;
struct node *rear;
void insert();
void delete();
void display();
void main ()
{
    int choice;
    while(choice != 4)
    {
```

```
printf("\n*****Main
Menu*****\n");

printf("\n=====
=====
\n");

printf("\n1.insert an element\n2.Delete an
element\n3.Display the queue\n4.Exit\n");
printf("\nEnter your choice ?");
scanf("%d",& choice);
switch(choice)
{
    case 1:
        insert();
        break;
    case 2:
        delete();
        break;
    case 3:
        display();
        break;
    case 4:
        exit(0);
        break;
    default:
        printf("\nEnter valid choice??\n");
}
}
}
```

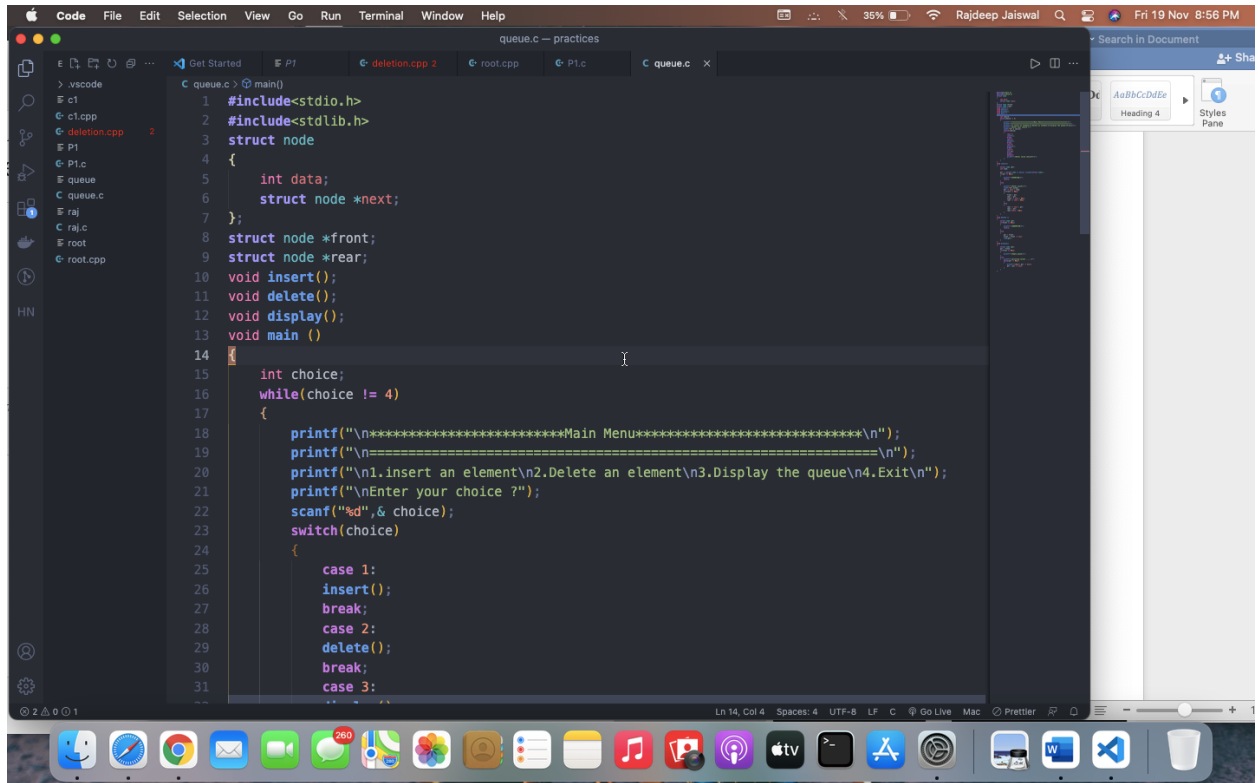
```
void insert()
{
    struct node *ptr;
    int item;

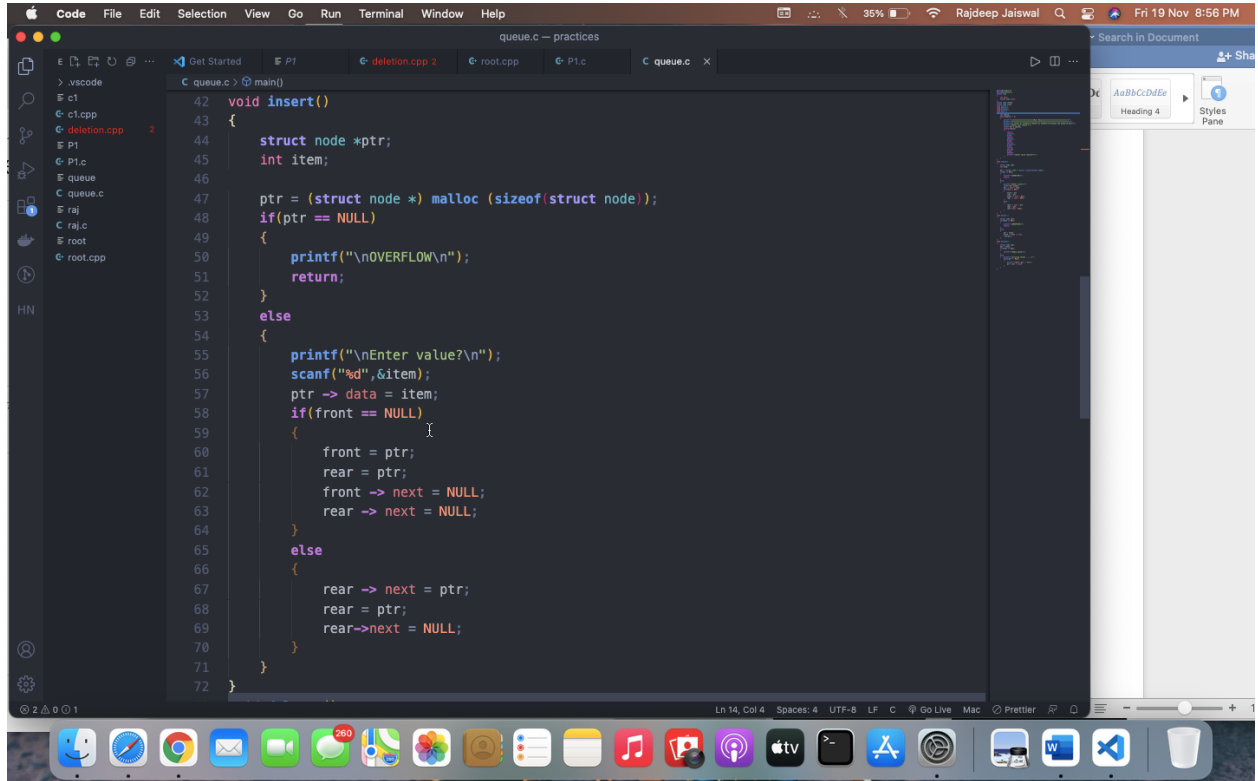
    ptr = (struct node *) malloc (sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW\n");
        return;
    }
    else
    {
        printf("\nEnter value?\n");
        scanf("%d",&item);
        ptr -> data = item;
        if(front == NULL)
        {
            front = ptr;
            rear = ptr;
            front -> next = NULL;
            rear -> next = NULL;
        }
        else
        {
            rear -> next = ptr;
            rear = ptr;
            rear->next = NULL;
        }
    }
}
```

```
    }  
  }  
}  
void delete ()  
{  
    struct node *ptr;  
    if(front == NULL)  
    {  
        printf("\nUNDERFLOW\n");  
        return;  
    }  
    else  
    {  
        ptr = front;  
        front = front -> next;  
        free(ptr);  
    }  
}  
void display()  
{  
    struct node *ptr;  
    ptr = front;  
    if(front == NULL)  
    {  
        printf("\nEmpty queue\n");  
    }  
    else  
    { printf("\nprinting values ..... \n");
```

```
while(ptr != NULL)
{
    printf("\n%d\n",ptr -> data);
    ptr = ptr -> next;
}
}
```

**CODE IN COMPILER-**





```
queue.c - practices
C queue.c > main()
42 void insert()
43 {
44     struct node *ptr;
45     int item;
46
47     ptr = (struct node *) malloc (sizeof(struct node));
48     if(ptr == NULL)
49     {
50         printf("\nOVERFLOW\n");
51         return;
52     }
53     else
54     {
55         printf("\nEnter value?\n");
56         scanf("%d",&item);
57         ptr -> data = item;
58         if(front == NULL)
59         {
60             front = ptr;
61             rear = ptr;
62             front -> next = NULL;
63             rear -> next = NULL;
64         }
65         else
66         {
67             rear -> next = ptr;
68             rear = ptr;
69             rear->next = NULL;
70         }
71     }
72 }
```

```
queue.c -- practices
C: queue.c > main()
73 void delete ()
74 {
75     struct node *ptr;
76     if(front == NULL)
77     {
78         printf("\nUNDERFLOW\n");
79         return;
80     }
81     else
82     {
83         ptr = front;
84         front = front -> next;
85         free(ptr);
86     }
87 }
88 void display()
89 {
90     struct node *ptr;
91     ptr = front;
92     if(front == NULL)
93     {
94         printf("\nEmpty queue\n");
95     }
96     else
97     {
98         printf("\nprinting values ..... \n");
99         while(ptr != NULL)
100         {
101             printf("\n%d\n", ptr -> data);
102             ptr = ptr -> next;
103         }
104     }
105 }
```

## OUTPUT –

```
*****Main Menu*****
=====
1.Insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice 71
Enter value?
5
*****Main Menu*****
=====
1.insert an element
2.Delete an element
3.Display the queue
4.Exit
Enter your choice 73
printing values .....
5
*****Main Menu*****
=====
```

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			