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

struct PQueue
{
    char n[4];
    int pr;
    struct PQueue *next;
} *front = NULL, *rear = NULL;

int count = 0;

void enqueue()
{
    struct PQueue *temp = malloc(sizeof(struct PQueue));
    if (temp == NULL)
        printf("Heap Overflow\n");
    else
    {
        printf("Enter the String : ");
        scanf("%s",temp->n);
        printf("Priority : ");
        scanf("%d",&temp->pr);
        temp->next = NULL;
        if (front == NULL || temp->pr < front->pr)
        {
            temp->next = front;
            front = temp;
        }
        else
        {
            struct PQueue *p = front;
            while (p->next != NULL && p->next->pr < temp->pr)
                p = p->next;
            temp->next = p->next;
            p->next = temp;
        }
        count++;
    }
}
void dequeue()
{
    if (front == NULL)
        printf("Queue Underflow\n");
```

```
    else
    {
        struct PQueue *temp = front;
        front = front->next;
        printf("Deleted Element : %s\n",temp->n);
        free(temp);
        count--;
    }
}
void Display()
{
    if (front == NULL)
        printf("Queue is Empty\n");
    else
    {
        struct PQueue *temp = front;
        printf("String\tPriority\n");
        while (temp != NULL)
        {
            printf("%s\t%d\n",temp->n,temp->pr);
            temp = temp->next;
        }
    }
}
int isEmpty()
{
    if (front == NULL)
        return 1;
    else
        return 0;
}
int main()
{
    int choice;
    printf("\n1. Enqueue\n2. Dequeue\n3. Front and Rear Element \n4.
Isempty\n5. Total no of element\n6. Display\n7. Exit\n");
    while (1)
    {
        printf("Enter the choice: ");
        scanf("%d", &choice);
        getchar();
        switch (choice)
        {
        case 1:
            enqueue();
            Display();
```

```
            break;
        case 2:
            dequeue();
            Display();
            break;
        case 3:
            printf("Front Element -> %s\n",front->n);
            struct PQueue *temp = front;
            while (temp->next != NULL)
            {
                temp = temp->next;
            }
            rear = temp;
            printf("Rear Element -> %s\n",rear->n);
            break;
        case 4:
            printf("%d\n", isEmpty());
            break;
        case 5:
            printf("Total number of elements -> %d\n", count);
            break;
        case 6:
            Display();
            break;
        case 7:
            printf("Exiting...");
            exit(0);
            break;
        }
    }
    return 0;
}
```

OUTPUT:

```
1. Enqueue
2. Dequeue
3. Front and Rear Element
4. Isempty
5. Total no of element
6. Display
7. Exit
Enter the choice: 1
Enter the String : abc
Priority : 1
String  Priority
abc     1
Enter the choice: 1
Enter the String : aka
Priority : 2
String  Priority
abc     1
aka     2
Enter the choice: 1
Enter the String : xyz
Priority : 0
String  Priority
xyz     0
abc     1
aka     2
Enter the choice: 2
Deleted Element : xyz
String  Priority
abc     1
aka     2
```

```
String  Priority
abc     1
aka     2
xyz     5
Enter the choice: 1
Enter the String : wvx
Priority : 3
String  Priority
abc     1
aka     2
wvx     3
xyz     5
Enter the choice: 3
Front Element -> abc
Rear Element -> xyz
Enter the choice: 4
0
Enter the choice: 5
Total number of elements -> 4
Enter the choice: 6
String  Priority
abc     1
aka     2
wvx     3
xyz     5
Enter the choice: 7
Exiting...
PS C:\Users\aadil\Desktop\CSE\dsalab>
```