Object Oriented Programming Lab
KCA 253 : Session 2020-21

Experiment -10

Write a program to implement queue using linked list

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

      printf("\n1.insert an element");
printf("\n2.Delete an element");
printf("\n3.peek");
printf("\n4.Exit\n");         printf("\nEnter
```

```
your choice ");          scanf("%d",&
choice);

      switch(choice)
      {
         case 1:
insert();
break;
case 2:
delete();
break;
case 3:
peek();      break;
case 4:
exit(0);
break;
default:
         printf("\nEnter wrong 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 peek()
{
   struct node *ptr;
   ptr = front;
   if(front == NULL)
   {
      printf("\nEmpty queue\n");
   }
   else
   {
      while(ptr != NULL)
      {
```

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

Output:

```
1.insert an element
2.Delete an element
3.peek
4.Exit

Enter your choice       1

Enter value
23

1.insert an element
2.Delete an element
3.peek
4.Exit

Enter your choice 1

Enter value
46

1.insert an element
2.Delete an element
3.peek
4.Exit

Enter your choice 3

23

46
```