Program of linear queue

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#Define QUE_SIZE 3
int item, front=0, rear=-1, q[10];
void insertrear()
{
 if (rear==QUE_SIZE-1)
{ printf("queue overflow\n");
  return;
}
  rear=rear+1;
  q[rear]=item;
}
int deletefront()
{ if(front>rear)
{
  front=0;
  rear=-1;
  return -1;
 }
 return q[front++];
}
void displayQ()
{
 int i;
 if (front>rear)
{ printf("queue is empty\n");
 return; }
```

```
printf("contents of queue\n");
for (i=front; i<=rear; i++)
{
printf("%d\n", q[i]);
}
}
void main()
{
int choice();
clrscr();
for(;;)
{
printf("\n1: insertrear\n2: deletefront\n3: display
\n4: exit\n");
printf("Enter the choice\n");
Scanf("%d", &choice);
Switch (choice)
{ case 1: printf("Enter the item to be inserted\n");
Scanf("%d", &item);
insertrear();
break;
case 2; item=deletefront();
if (item==-1)
printf("queue is empty\n");
else
printf("item deleted =%d\n", item);
break;
case 3: displayQ(); break;
default: exit(0);
```

```c
#include <stdio.h>
#include<stdlib.h>
#define QUE_SIZE 3
int item,front=0,rear=-1,q[10];
void insertrear()
 {
     if(rear==QUE_SIZE-1)
     {
         printf("Queue overflow\n");
         return;
     }
     rear=rear+1;
     q[rear]=item;
 }
 int deletefront(){
     if(front>rear)
     {
         front=0;
         rear=-1;
         return -1;
     }
     return q[front++];
 }
 void displayQ()
 {
     int i;
     if(front>rear){
        printf("Queue is empty");
        return;
     }
     else
     printf("contents of queue\n");
     for(i=front;i<=rear;i++){
        printf("%d\n",q[i]);
     }
 }
void main()
{
    int choice;
    for(;;){
        printf("\n 1:insert rear\n 2: delete front\n 3:display\n 4:exit\n");
        printf("Enter the choice\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:printf("Enter the item to be inserted\n");
                   scanf("%d",&item);
                   insertrear();
            break;
            case 2:item=deletefront();
                   if(item==-1)
                   printf("queue is empty\n");
                   else
                   printf("item deleted=%d\n",item);
             break;
             case 3:displayQ();
             break;
             default:exit(0);
        }
    }

}
```

```
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
4

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
3

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
3

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
3
contents of queue
4
3
3
```

```
 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
5

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
4

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
3

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
2
Queue overflow
```

```
 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
1
Enter the item to be inserted
5

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
2
item deleted=5

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
2
queue is empty

 1:insert rear
 2: delete front
 3:display
 4:exit
Enter the choice
```