linearqueues.c > displayq()

```c
#include<stdio.h>
#include<stdlib.h>
#define que_size 3
int item,front=0,rear=-1,q[que_size];
void insertrear(){
    if(rear==que_size-1)
    {
        printf("queue overflow");
        return;
    }

    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;
    }
    printf("contents of queue \n");
    for(i=front;i<=rear;i++)
    {
        printf("%d\n",q[i]);
    }
}
int main(){
    int choice;
    for(;;)
```

linearqueues.c > displayq()

```c
        return;
    }
    printf("contents of queue \n");
    for(i=front;i<=rear;i++)
    {
        printf("%d\n",q[i]);
    }
}
int main(){
    int choice;
    for(;;)
    {
        printf("1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
        printf("Enter the choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: printf("Enter the item\n");
                    scanf("%d",&item);
                    insertrear();
                   break;
            case 2:item=deletefront();
                   if(item==-1)
                   printf("queue is empty\n");
                   else
                   printf("item deleted is %d \n",item);
                   break;
            case 3:displayq();
                   break;
            default:exit(0);
        }
    }
}
```

linearqueues.c > displayq()

```
27          return;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Code

```
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item
10
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item
20
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item
30
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 10
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
20
30
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice :
```

Simple Queue

```
#include <stdio.h>
#include <conio.h>
# define QUE_SIZE 5
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 choic;
  for (;;)
  { printf("\n1: insert rear \n2: delete front \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);
               insert rear();
                break;
      case 2: item = delete front();
                 if (item == -1)
               printf("queue is empty \n");
                else
               printf("item deleted = %d \n", item);
                 break;
        case 3: display-Q();
                  break;

          case 4: exit;

          }
}
```