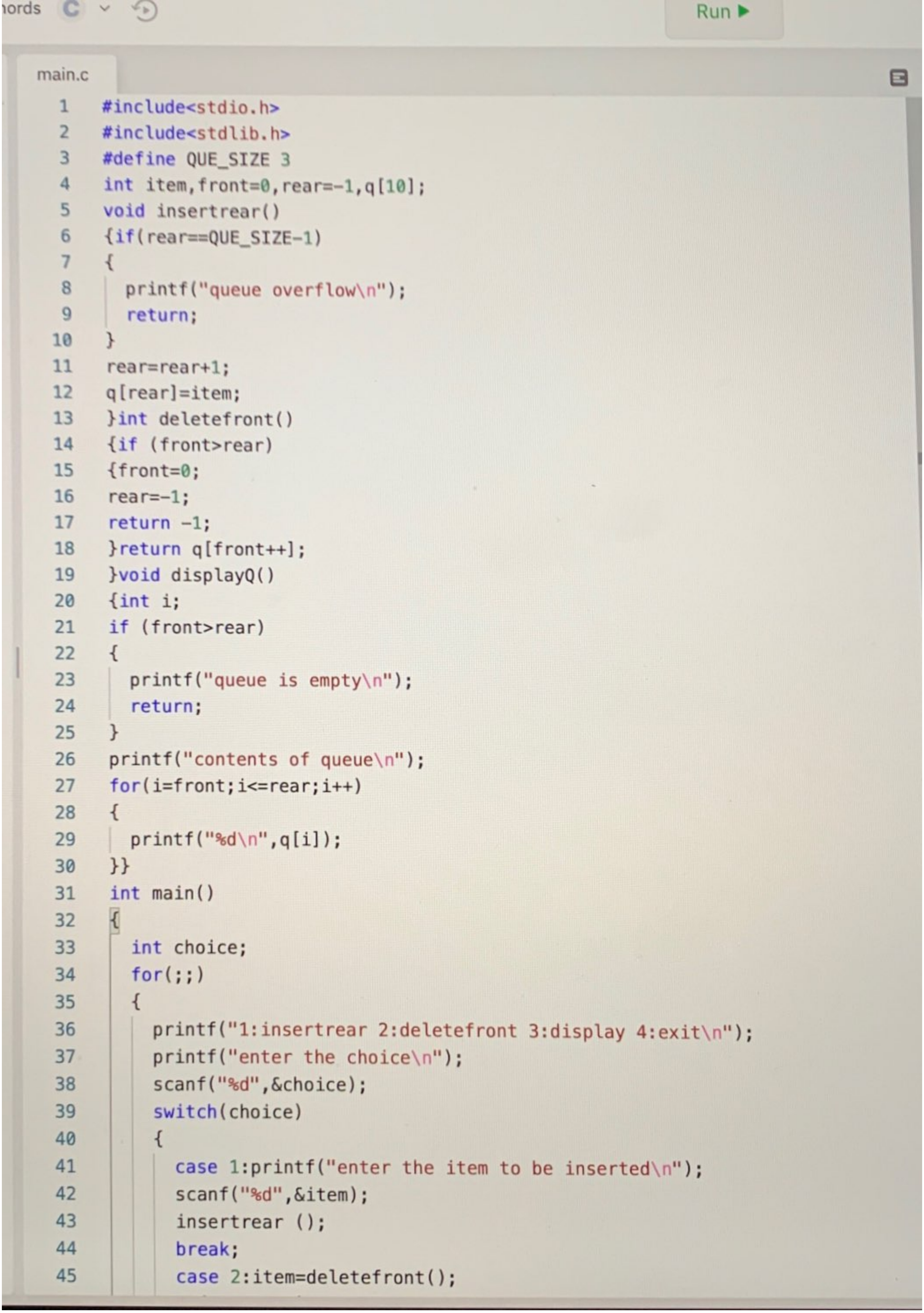
S.R.POOJA
1BM19CS135

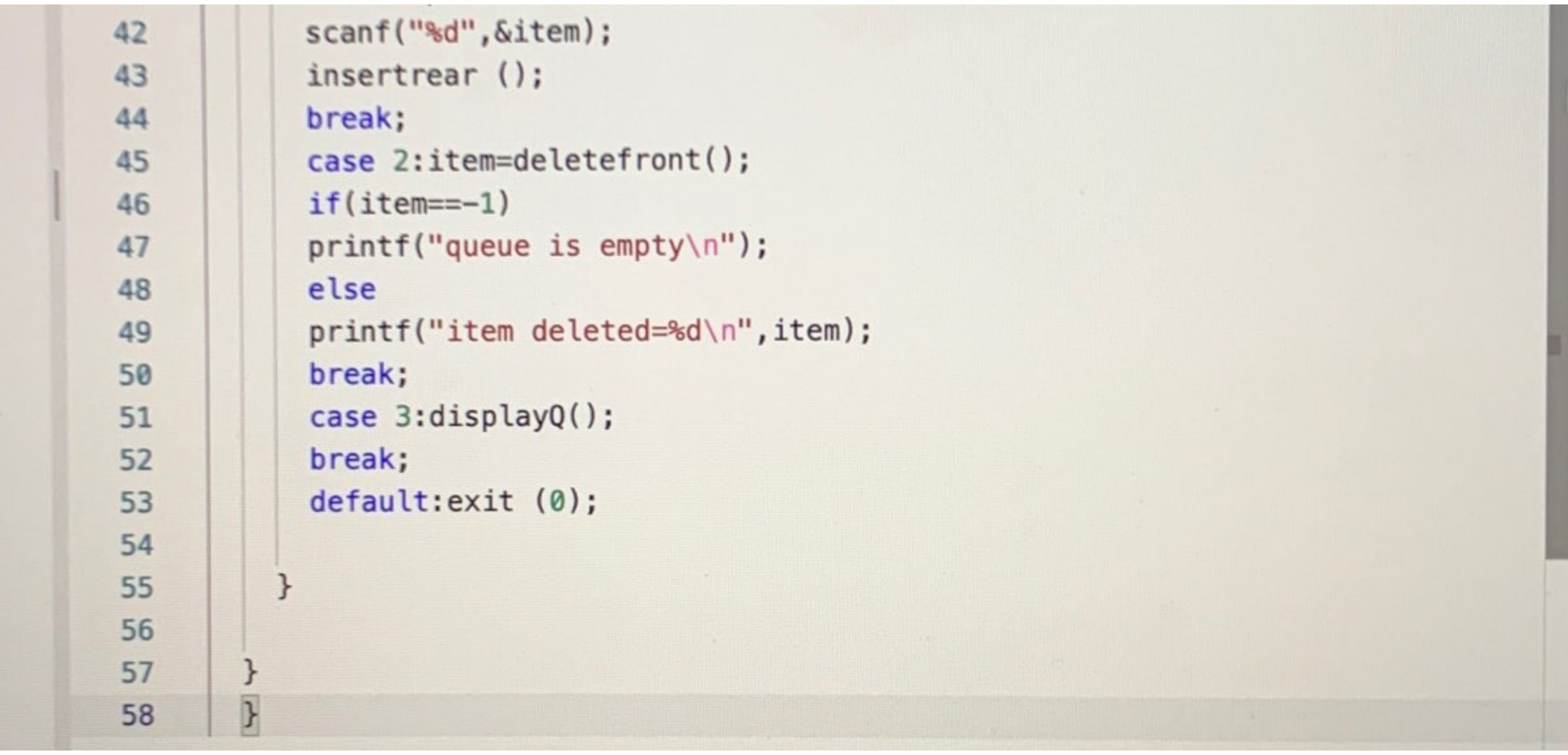
QUEUE (ORDINARY)

```
#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 display()
{ 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]);
}
}
```

```
int main()
{
int choice;
for (;;)
{
printf("1: insert rear  2: delete front 3: display 4: exit \n");
printf("enter the choice \n");
scanf("%d", &choice);
switch (choice)
{
case 1: printf(" enter the ~~int~~ 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);
}
}
}
```



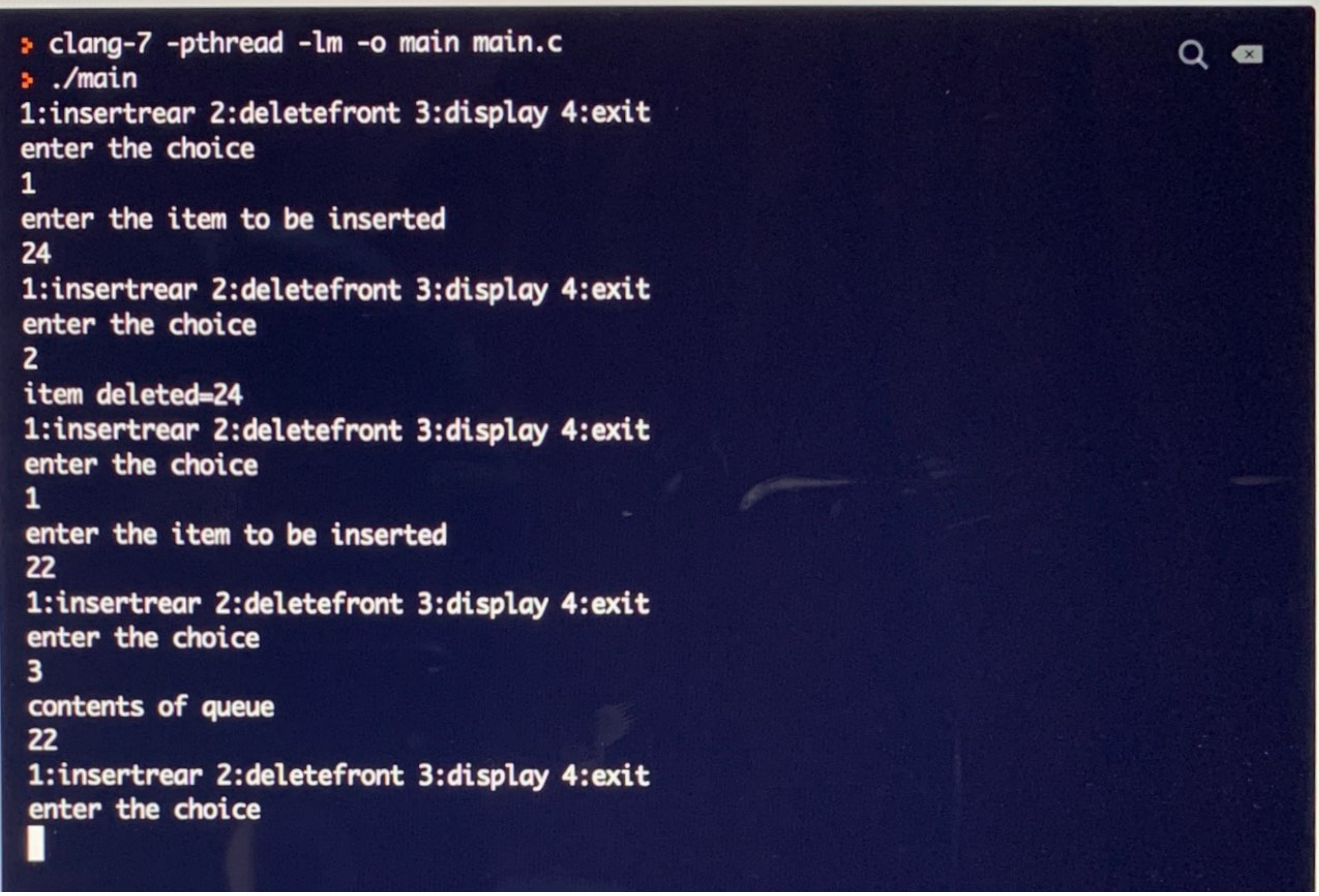
```c
#include<stdio.h>
#include<stdlib.h>
#define QUE_SIZE 3
int item,front=0,rear=-1,q[10];
void insertrear()
{if(rear==QUE_SIZE-1)
{
  printf("queue overflow\n");
  return;
}
rear=rear+1;
q[rear]=item;
}int deletefront()
{if (front>rear)
{front=0;
rear=-1;
return -1;
}return q[front++];
}void displayQ()
{int i;
if (front>rear)
{
  printf("queue is empty\n");
  return;
}
printf("contents of queue\n");
for(i=front;i<=rear;i++)
{
  printf("%d\n",q[i]);
}}
int main()
{
  int choice;
  for(;;)
  {
    printf("1:insertrear 2:deletefront 3:display 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();
```



```
        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);

    }

}
}
```



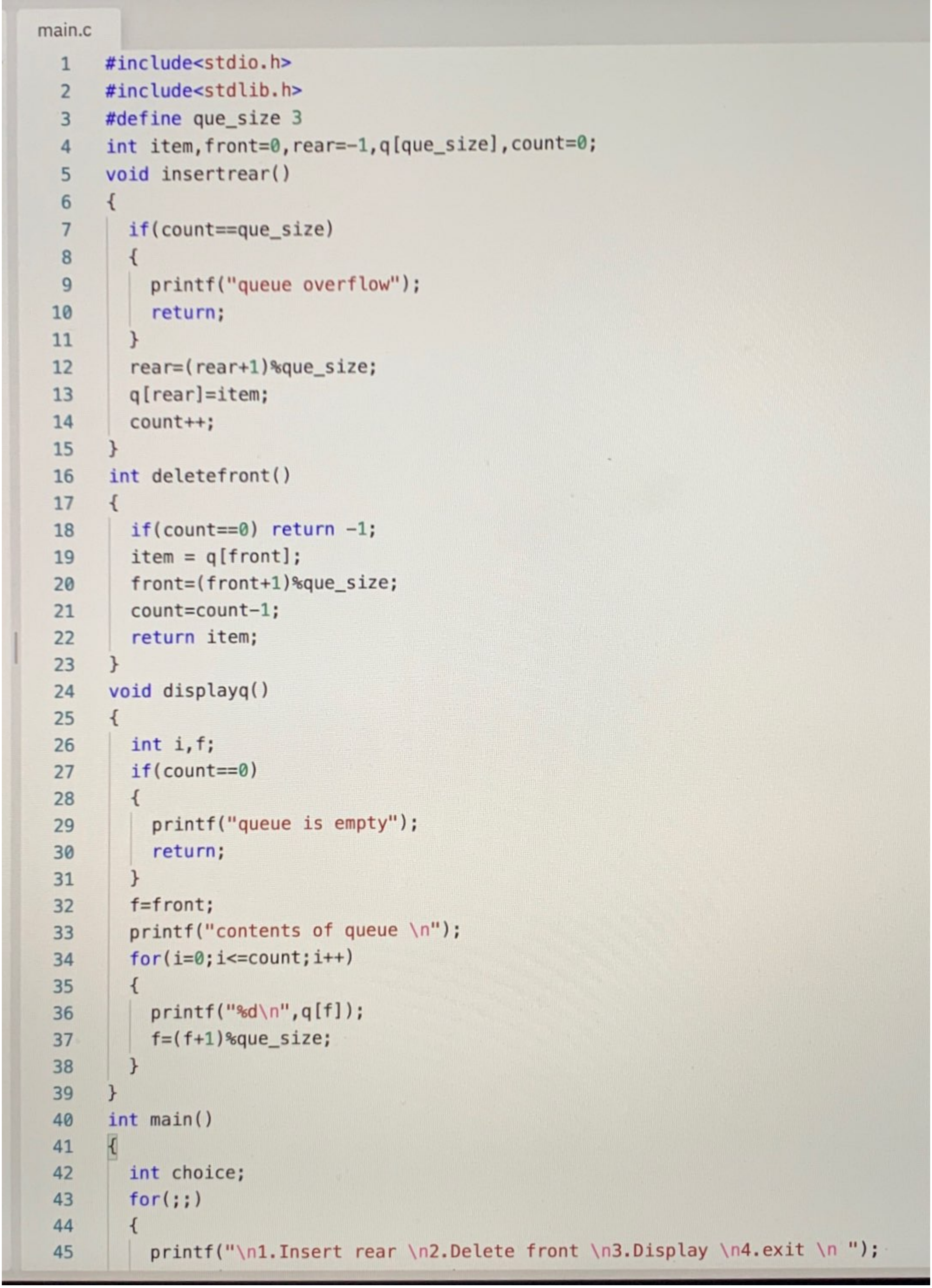
```
> clang-7 -pthread -lm -o main main.c
> ./main
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
24
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
2
item deleted=24
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
22
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
3
contents of queue
22
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
```

S.R.POOJA
1BM19CS135

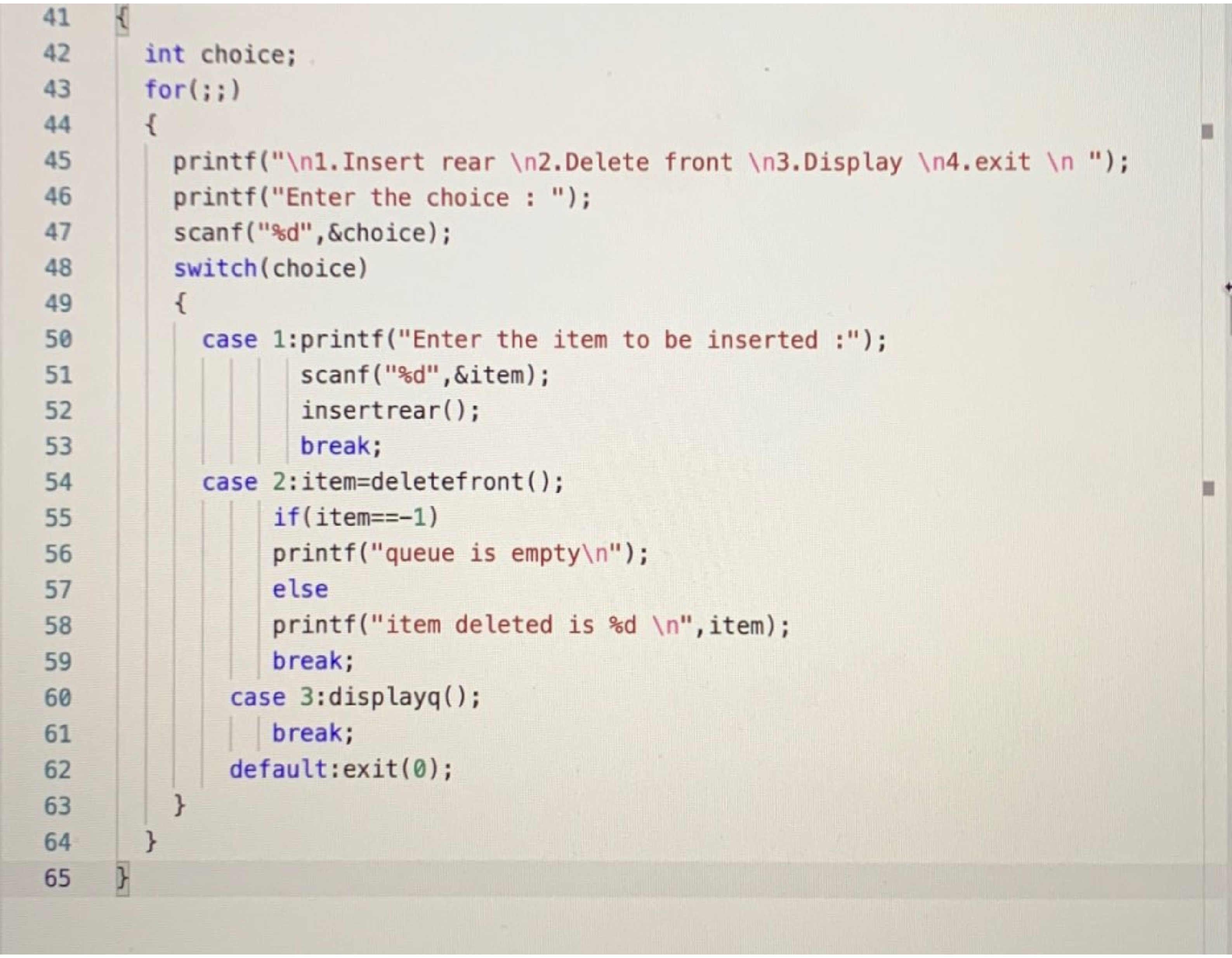
CIRCULAR QUEUE

```
#include<stdio.h>
#include <stdlib.h>
#define que-size 3
int item, front = 0, rear = -1, q[que-size], count=0;
void insertrear()
{
if (count =que-size)
{
printf("queue overflow");
return ;
}
rear=(rear+1) % que-size;
~~count = count-1;~~
q[rear] =item;
count++;
}
int deletefront()
{
if (count==0) return -1;
item = q[front];
front=(front+1) % que-size;
count=count-1;
return item;
}
void displayq()
{
int i,f;
if (count==0)
 {
```

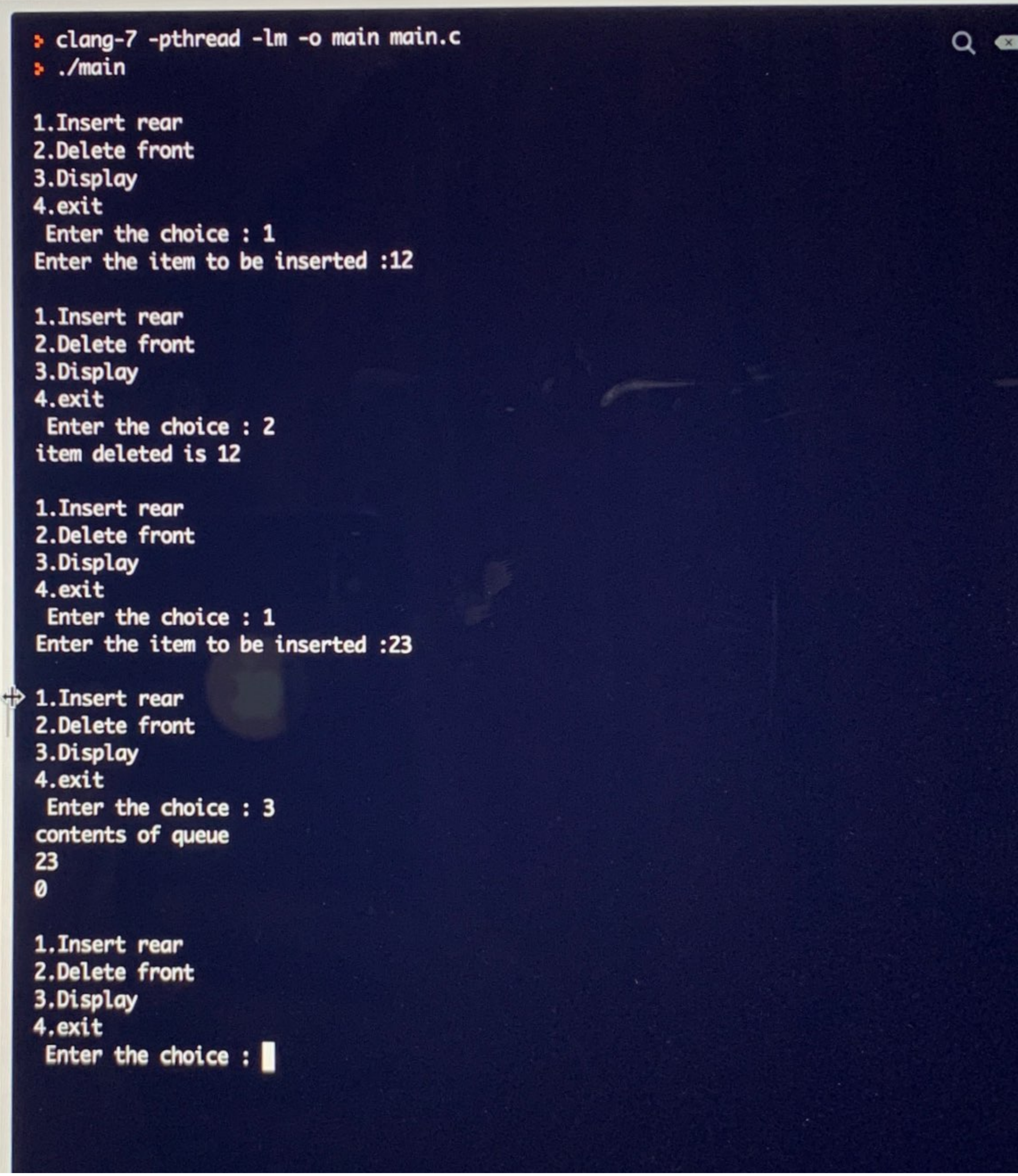
```
printf("queue is empty");
return;
}
f=front;
printf("contents of queue \n");
for(i=0; i<=count; i++)
{
printf("%d\n", q[f]);
f=(f+1) % que_size;
}
}
int main()
{
int choice;
for(;;)
{
printf("\n 1.Insert rear \n 2.Delete front \n 3.Display \n4.exit\n")
printf("Enter the choice:");
scanf("%d", &choice);
switch(choice)
{
case 1: printf(Enter the item to be inserted:");
    scanf("%d", & item);
    insert rear();
    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);
 }
}
```



```c
#include<stdio.h>
#include<stdlib.h>
#define que_size 3
int item,front=0,rear=-1,q[que_size],count=0;
void insertrear()
{
  if(count==que_size)
  {
    printf("queue overflow");
    return;
  }
  rear=(rear+1)%que_size;
  q[rear]=item;
  count++;
}
int deletefront()
{
  if(count==0) return -1;
  item = q[front];
  front=(front+1)%que_size;
  count=count-1;
  return item;
}
void displayq()
{
  int i,f;
  if(count==0)
  {
    printf("queue is empty");
    return;
  }
  f=front;
  printf("contents of queue \n");
  for(i=0;i<=count;i++)
  {
    printf("%d\n",q[f]);
    f=(f+1)%que_size;
  }
}
int main()
{
  int choice;
  for(;;)
  {
    printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
```



```
{
  int choice;
  for(;;)
  {
    printf("\n1.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 to be inserted :");
             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);
    }
  }
}
```



```
> clang-7 -pthread -lm -o main main.c
> ./main

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :12

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 12

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :23

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
23
0

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : █
```