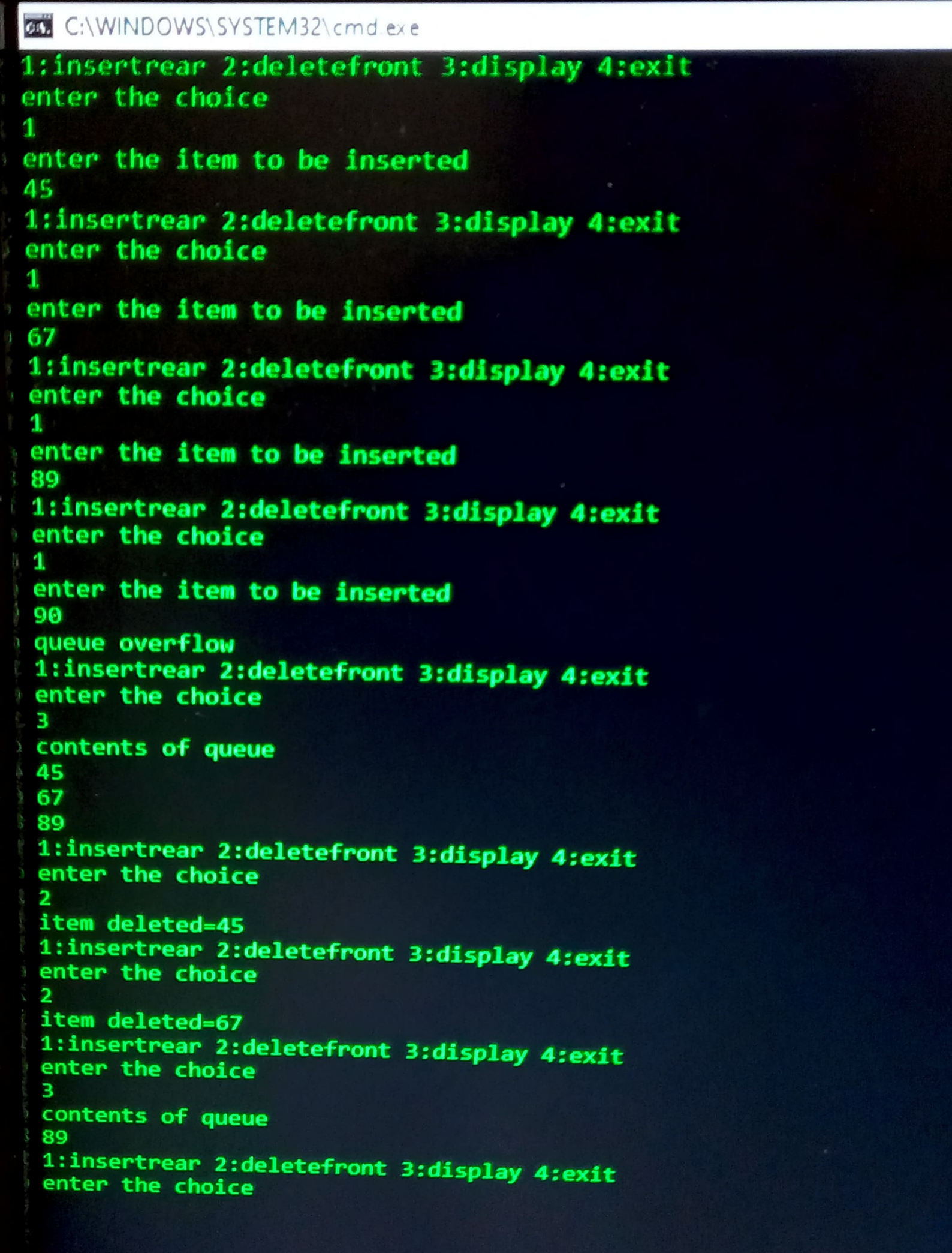
Program:

```
#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 0;
  }
  rear = rear+1;
  q[rear] = item;
}
```

```
int delfront()
{
  if (front > rear)
  {
    front = 0;
    rear = -1;
    return -1;
  }
  return q[front++];
}
void display()
{
  int i;
  if (front > rear)
  {
    printf("queue underflow");
    return;
  }
  printf("Contents of queue \n");
  for (i = front; i <= rear; i++)
  {
    printf("%d\n", q[i]);
  }
}
```

```
int main()
{
  int choice;
  print
   for(;;)
    {
    printf("\n 1: insert rear \n 2: delet fron
                3: display \n 4: exit\n");
    printf("enter the choice \n");
    scanf("%d", &choice);
    switch(choice)
    {
    case 1: printf(" enter the item t
                be inserted \n");
            scanf("%d", &item);
            insertrear();
            break;
    case 2: item = deletefron();
            if(item == -1)
            printf("que empty\n");
            else
            printf("item deleted =%
                    %d", item);
            break;
```

```
case 3: display();
        break;
default: exit(0);
}
}
```


C:\WINDOWS\SYSTEM32\cmd.exe
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
45
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
67
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
89
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
90
queue overflow
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
3
contents of queue
45
67
89
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
2
item deleted=45
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
2
item deleted=67
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
3
contents of queue
89
1:insertrear 2:deletefront 3:display 4:exit
enter the choice

Syntax

1
```
#include <stdio.h>
#include <stdlib.h>
#include
#define que_size 3

int item, front=0, rear=-1, q[que_size], count
void insertrear()
{
 if (count == que_size)
 {
  printf("Que overflow\n");
  return;
 }
rear = (rear+1) % que_size;
q[rear] = item;
 count++;
}
int deletefront()
{
   if (count==0) return 7;
   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\n");
    return;
  }
  f = front;
  printf ("Contents of queue\n");
  for (i = 1; i <= count; i++)
  {
    printf ("%d\n", q[f]);
    f = (f+1) % que_size;
  }
}
void main() {
  int choice;
  for(;;) {
    printf("\n 1. insert\n 2. delete
           \n 3. display 4. exit");
```

```
printf("enter the choice\n");
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("que is empty\n");
        else
        printf("item deleted = %d\n", item);
        break;
case 3: display();
        break;
default : exit(0);
}
}
}
```

C:\WINDOWS\SYSTEM32\cmd.exe

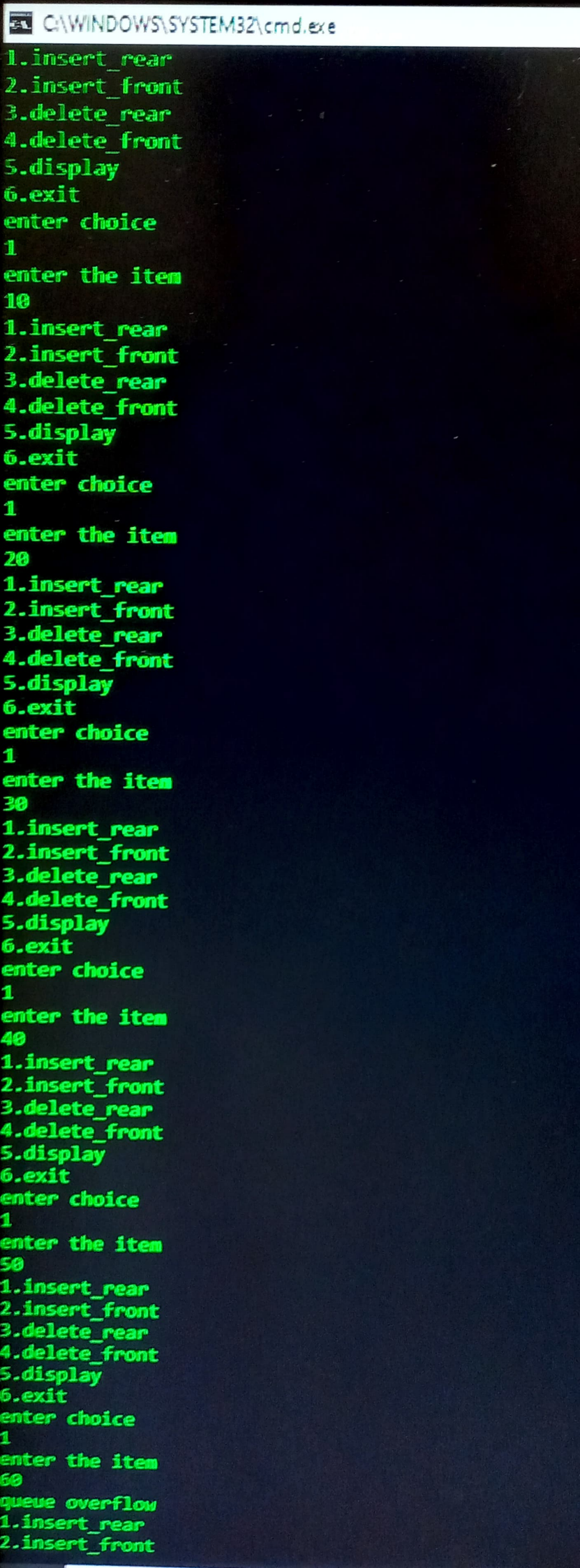
```
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :45

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :78

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :9

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
45
78
9

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 45

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
78
9

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

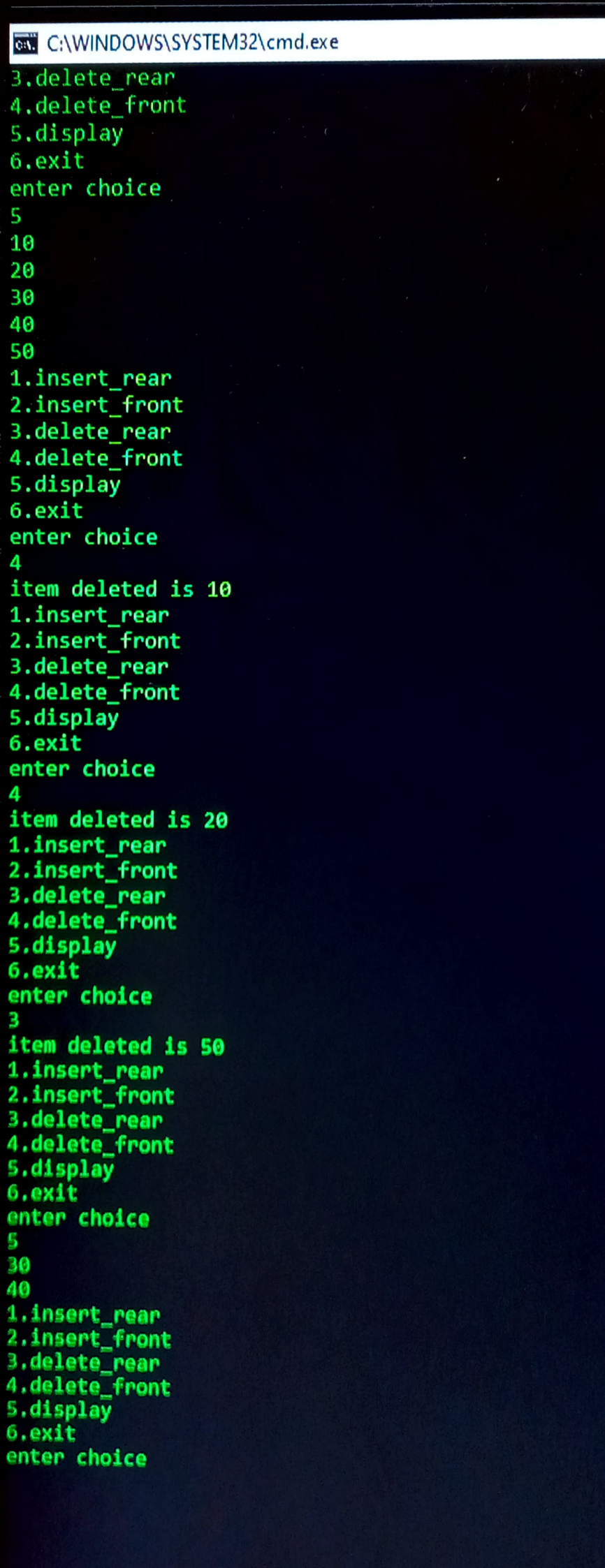
```
#include <stdio.h>
#include <conio.h>
#include <process.h>
int f = 0, r = -1, ch;
int item, q[10];
int isfull()
{
return (r == qsize-1) ? 1:0;
}
int isempty(int f, int r)
{
return (f > r) ? 1:0;
}
void insert_rear()
{
if (isfull())
{
printf("queue overflow\n");
return;
}
r = r+1;
q[r] = item;
}
```

```
void delet-front()
{
    if (isempty())
    {
        printf("queue empty\n");
        return;
    }
    else if ((f==0)&&
printf("item deleted is %d\n", q[f++]);
    if (f>r)
    {
        f=0
        r=-1;
    }
}
void display() {
    int i;
    if (isempty()) {
        printf("queue empty\n");
        return;
    }
    for(i=f; i<=r; i++)
        printf("%d\n", q[i]);
}
```

```
int main()
{
for (;;) {
printf("1. insert_rear \n 2. insert_front \n 3. d
    rear 4. delete_front \n 5. displa
        6. exit \n");
printf(" enter choice \n");
scanf ("%d", &ch);
switch(ch)
    {
    case 1: printf("enter item\n");
            scanf ("%d", &item);
            insert_rear();
            break;
    case 2: printf ("enter the item \n");
            scanf ("%d", &item);
            insert_front();
            break;
```

```
        case 3: delete_rear();
                break;
        case 4: delete_front();
                break;
        default: exit(0);
        }
    }
}
```


C:\WINDOWS\SYSTEM32\cmd.exe
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
60
queue overflow
1.insert_rear
2.insert_front


C:\WINDOWS\SYSTEM32\cmd.exe
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
10
20
30
40
50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
30
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice