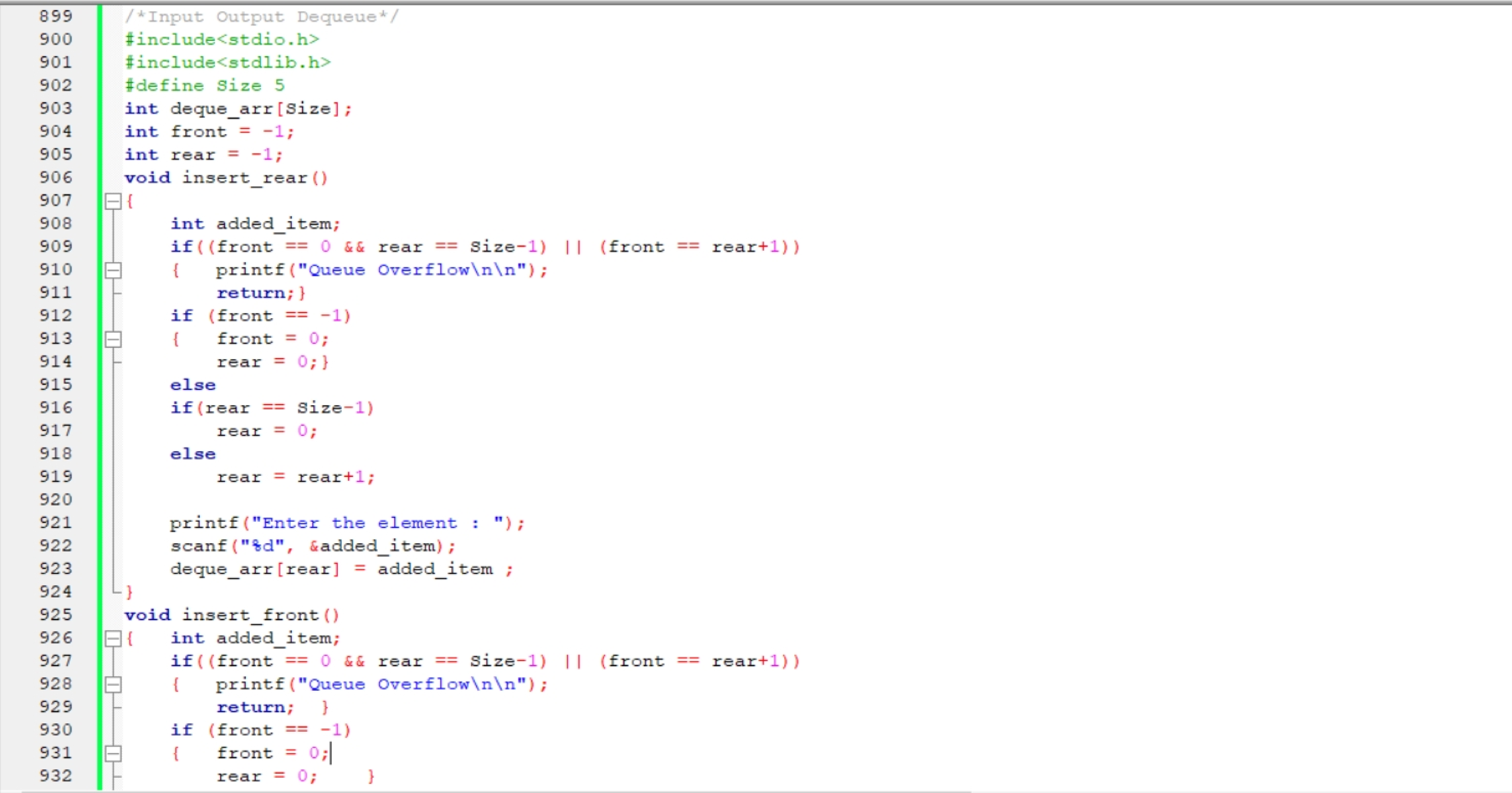
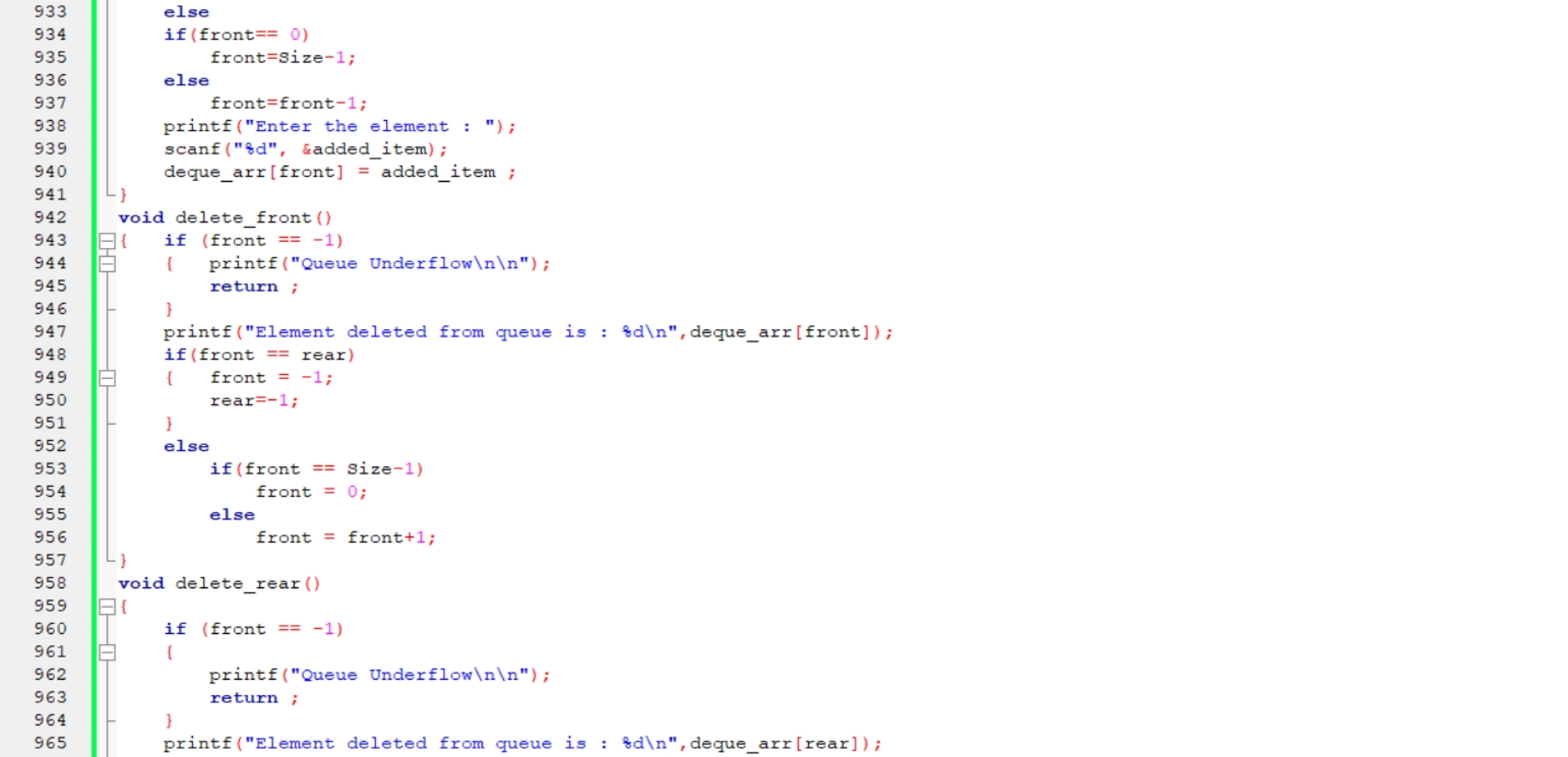
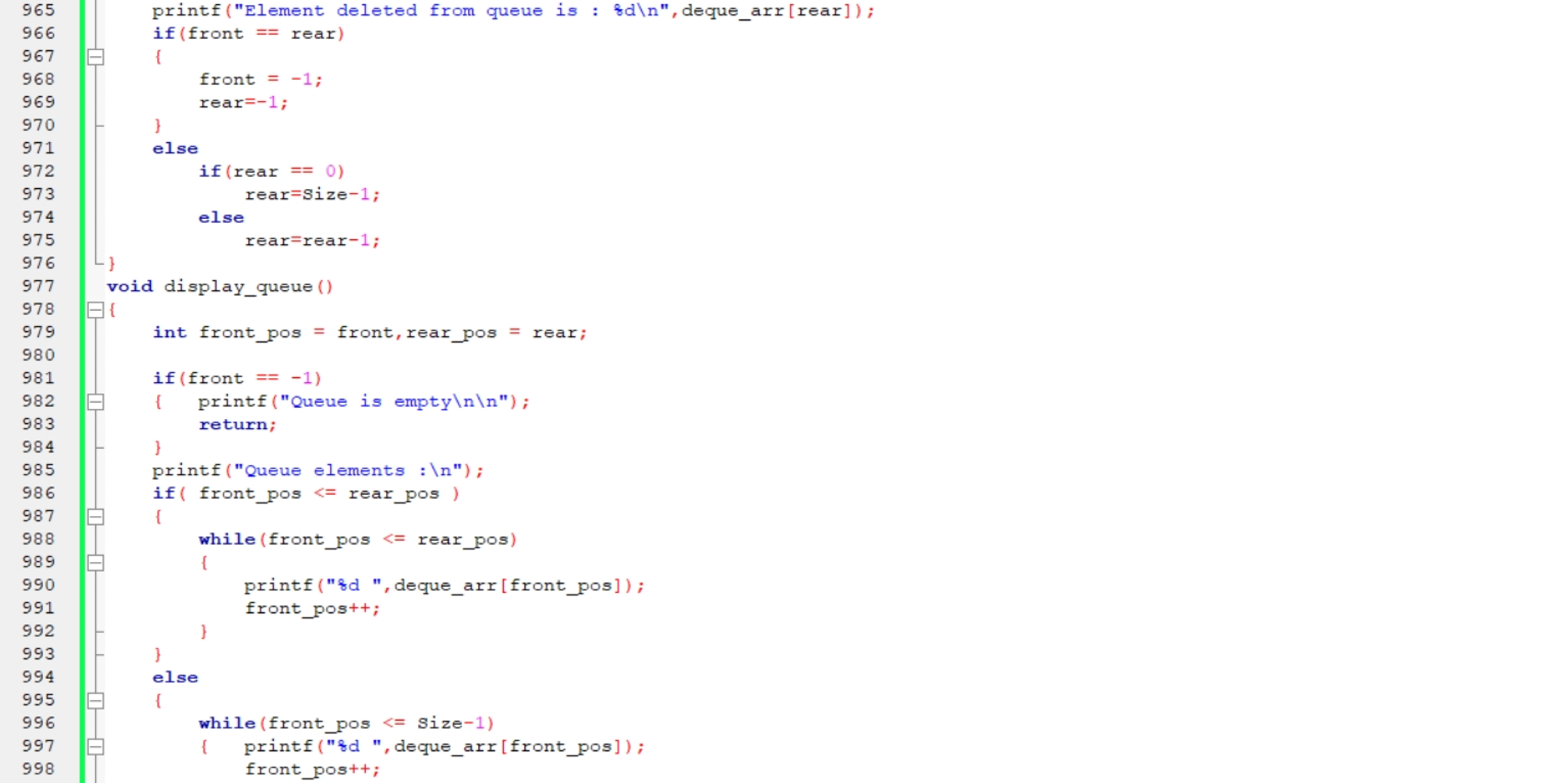


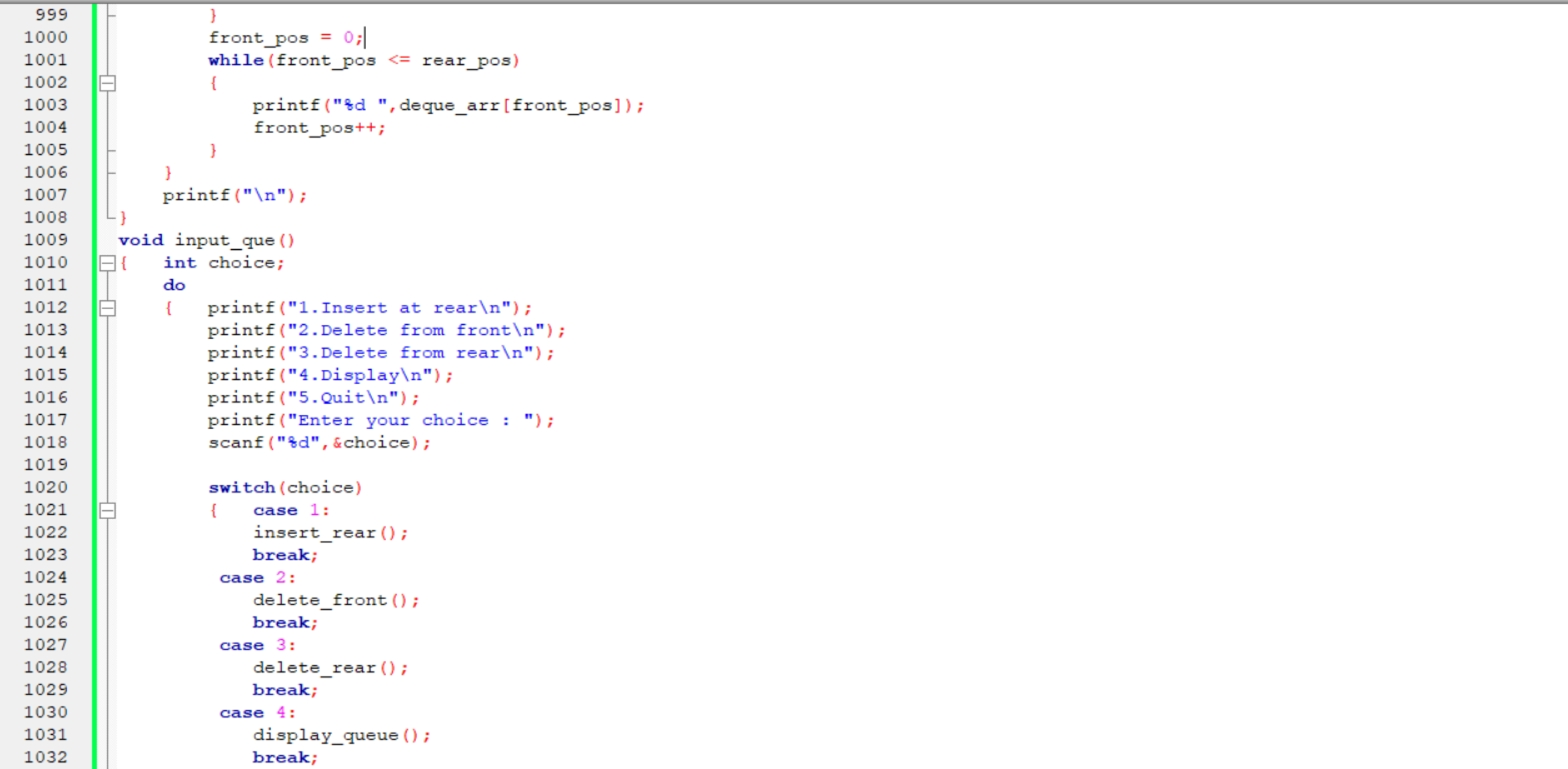
```c
/*Input Output Dequeue*/
#include<stdio.h>
#include<stdlib.h>
#define Size 5
int deque_arr[Size];
int front = -1;
int rear = -1;
void insert_rear()
{
    int added_item;
    if((front == 0 && rear == Size-1) || (front == rear+1))
    {   printf("Queue Overflow\n\n");
        return;}
    if (front == -1)
    {   front = 0;
        rear = 0;}
    else
    if(rear == Size-1)
        rear = 0;
    else
        rear = rear+1;

    printf("Enter the element : ");
    scanf("%d", &added_item);
    deque_arr[rear] = added_item ;
}
void insert_front()
{   int added_item;
    if((front == 0 && rear == Size-1) || (front == rear+1))
    {   printf("Queue Overflow\n\n");
        return;  }
    if (front == -1)
    {   front = 0;
        rear = 0;    }
```



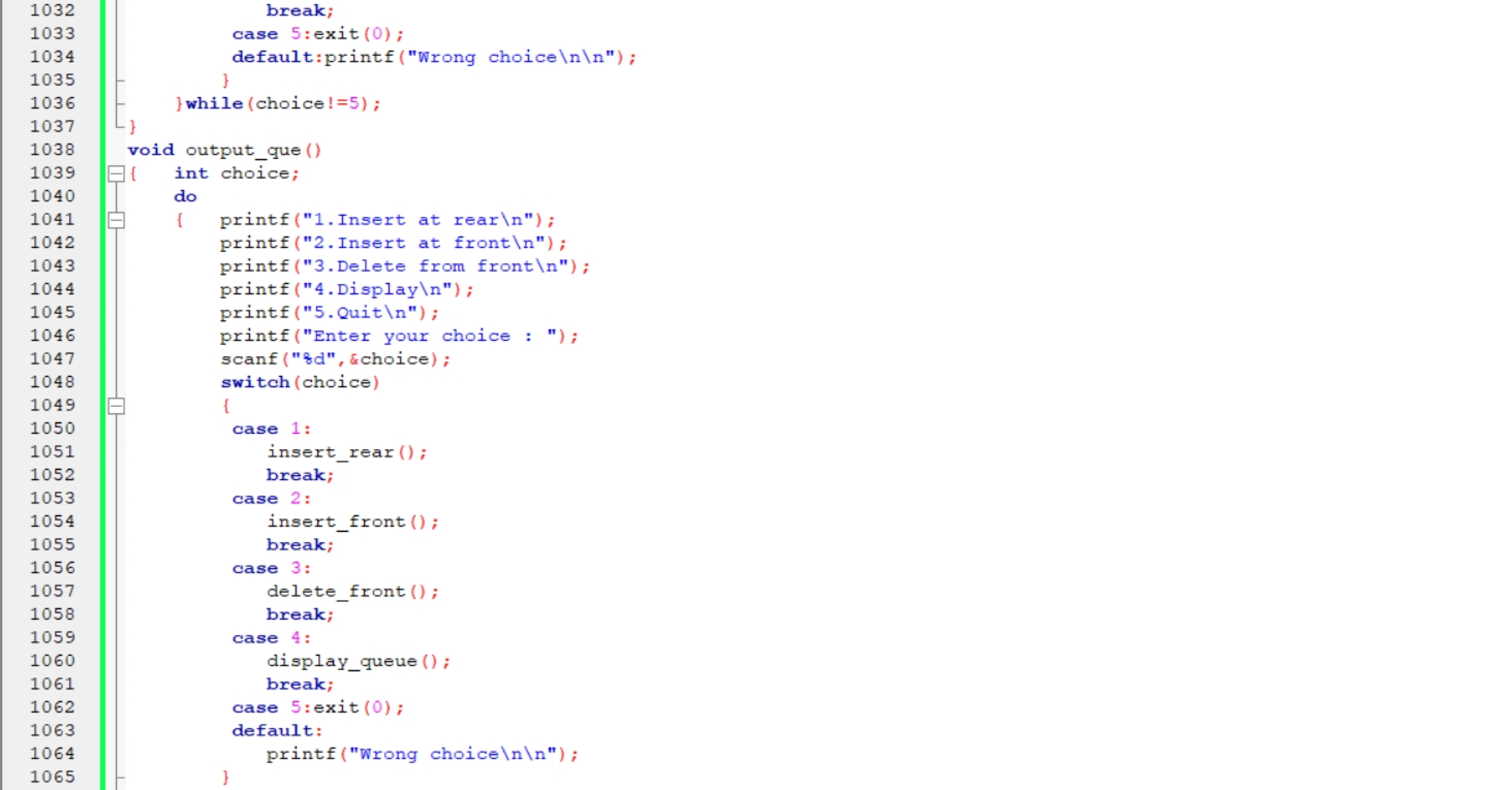
```
    else
    if(front== 0)
        front=Size-1;
    else
        front=front-1;
    printf("Enter the element : ");
    scanf("%d", &added_item);
    deque_arr[front] = added_item ;
}
void delete_front()
{   if (front == -1)
    {   printf("Queue Underflow\n\n");
        return ;
    }
    printf("Element deleted from queue is : %d\n",deque_arr[front]);
    if(front == rear)
    {   front = -1;
        rear=-1;
    }
    else
        if(front == Size-1)
            front = 0;
        else
            front = front+1;
}
void delete_rear()
{
    if (front == -1)
    {
        printf("Queue Underflow\n\n");
        return ;
    }
    printf("Element deleted from queue is : %d\n",deque_arr[rear]);
```



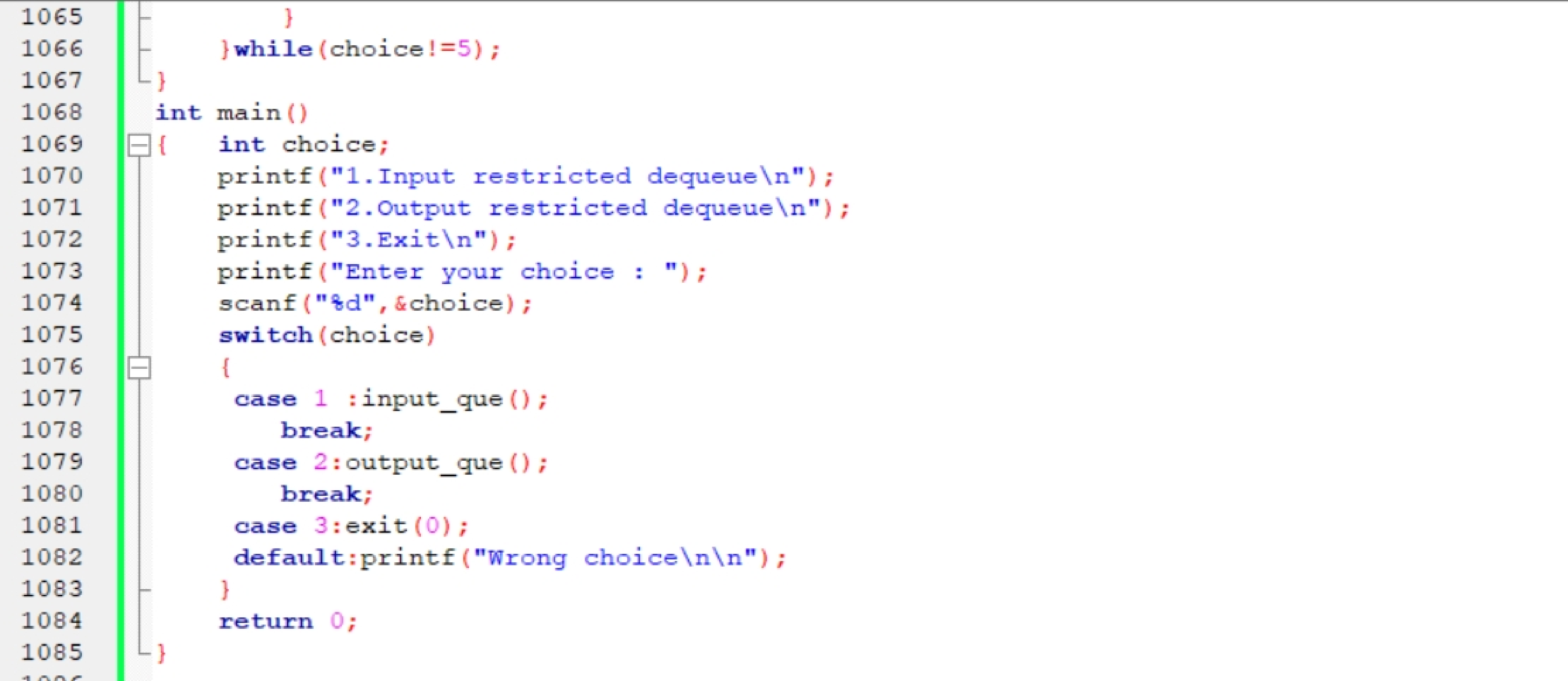
```c
    printf("Element deleted from queue is : %d\n",deque_arr[rear]);
    if(front == rear)
    {
        front = -1;
        rear=-1;
    }
    else
        if(rear == 0)
            rear=Size-1;
        else
            rear=rear-1;
}
void display_queue()
{
    int front_pos = front,rear_pos = rear;

    if(front == -1)
    {   printf("Queue is empty\n\n");
        return;
    }
    printf("Queue elements :\n");
    if( front_pos <= rear_pos )
    {
        while(front_pos <= rear_pos)
        {
            printf("%d ",deque_arr[front_pos]);
            front_pos++;
        }
    }
    else
    {
        while(front_pos <= Size-1)
        {   printf("%d ",deque_arr[front_pos]);
            front_pos++;
```



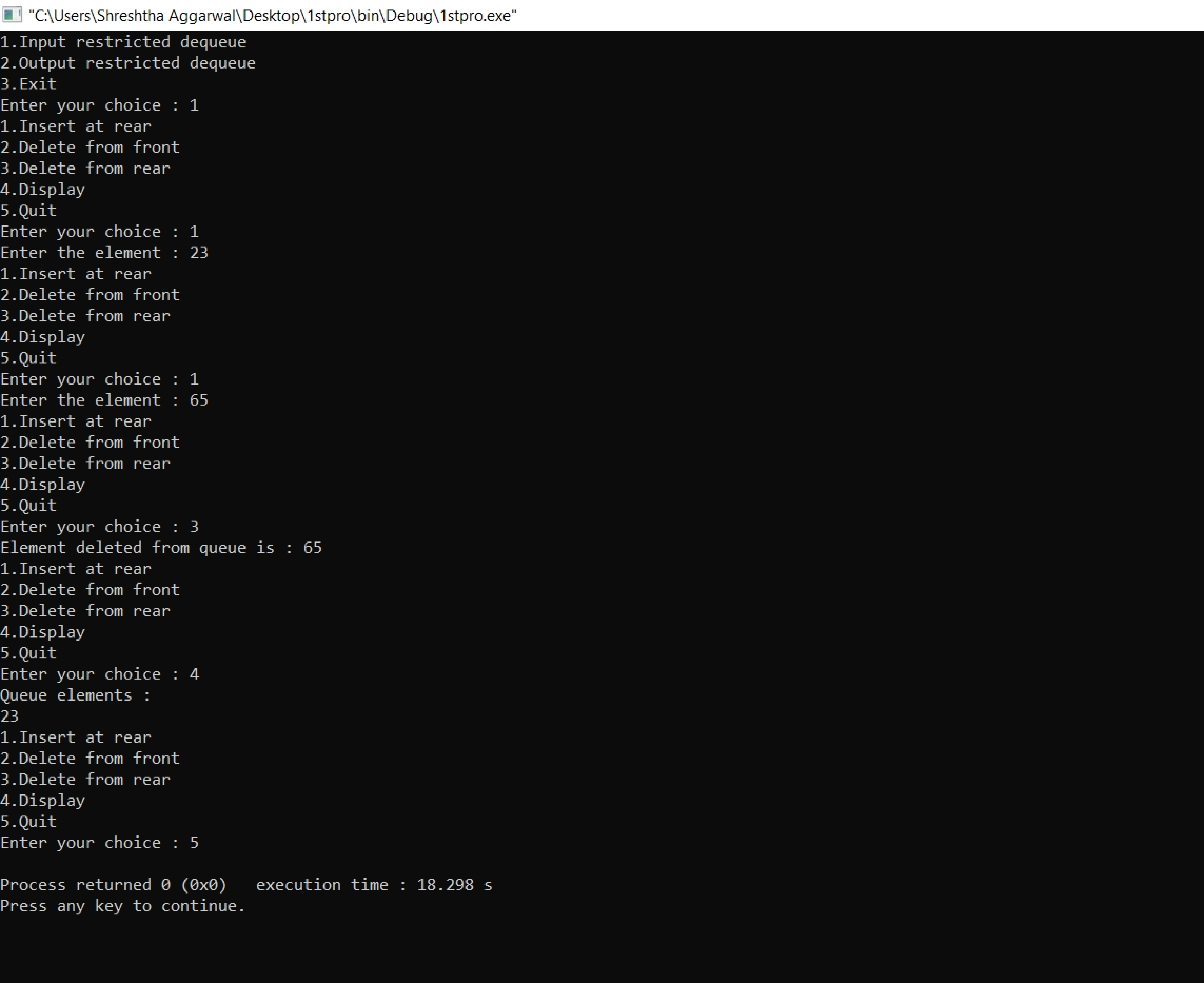
```
        }
        front_pos = 0;
        while(front_pos <= rear_pos)
        {
            printf("%d ",deque_arr[front_pos]);
            front_pos++;
        }
    }
    printf("\n");
}
void input_que()
{   int choice;
    do
    {   printf("1.Insert at rear\n");
        printf("2.Delete from front\n");
        printf("3.Delete from rear\n");
        printf("4.Display\n");
        printf("5.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);

        switch(choice)
        {   case 1:
            insert_rear();
            break;
         case 2:
            delete_front();
            break;
         case 3:
            delete_rear();
            break;
         case 4:
            display_queue();
            break;
```



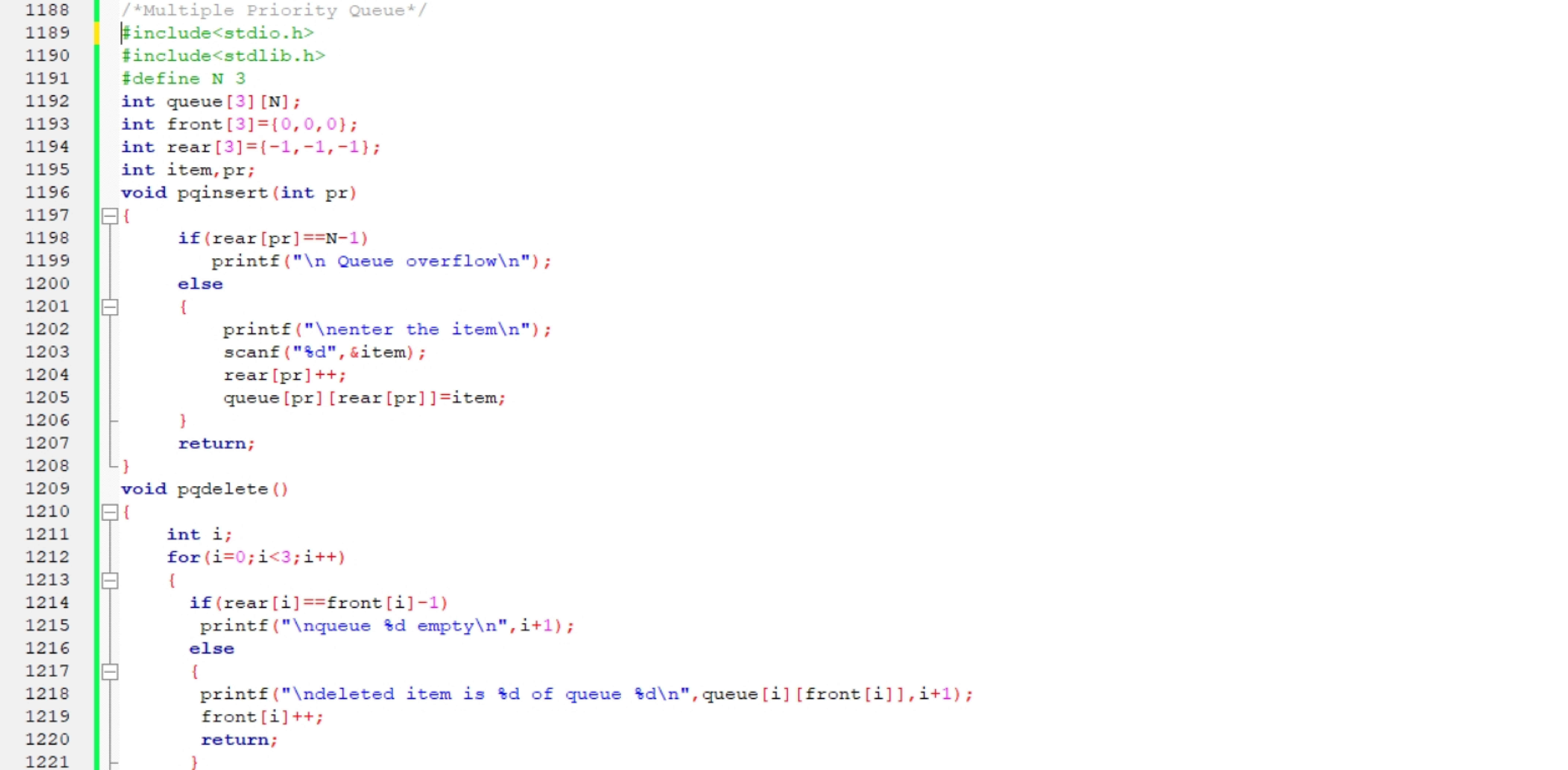
```
            break;
         case 5:exit(0);
         default:printf("Wrong choice\n\n");
        }
    }while(choice!=5);
}
void output_que()
{   int choice;
    do
    {   printf("1.Insert at rear\n");
        printf("2.Insert at front\n");
        printf("3.Delete from front\n");
        printf("4.Display\n");
        printf("5.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
         case 1:
            insert_rear();
            break;
         case 2:
            insert_front();
            break;
         case 3:
            delete_front();
            break;
         case 4:
            display_queue();
            break;
         case 5:exit(0);
         default:
            printf("Wrong choice\n\n");
        }
```



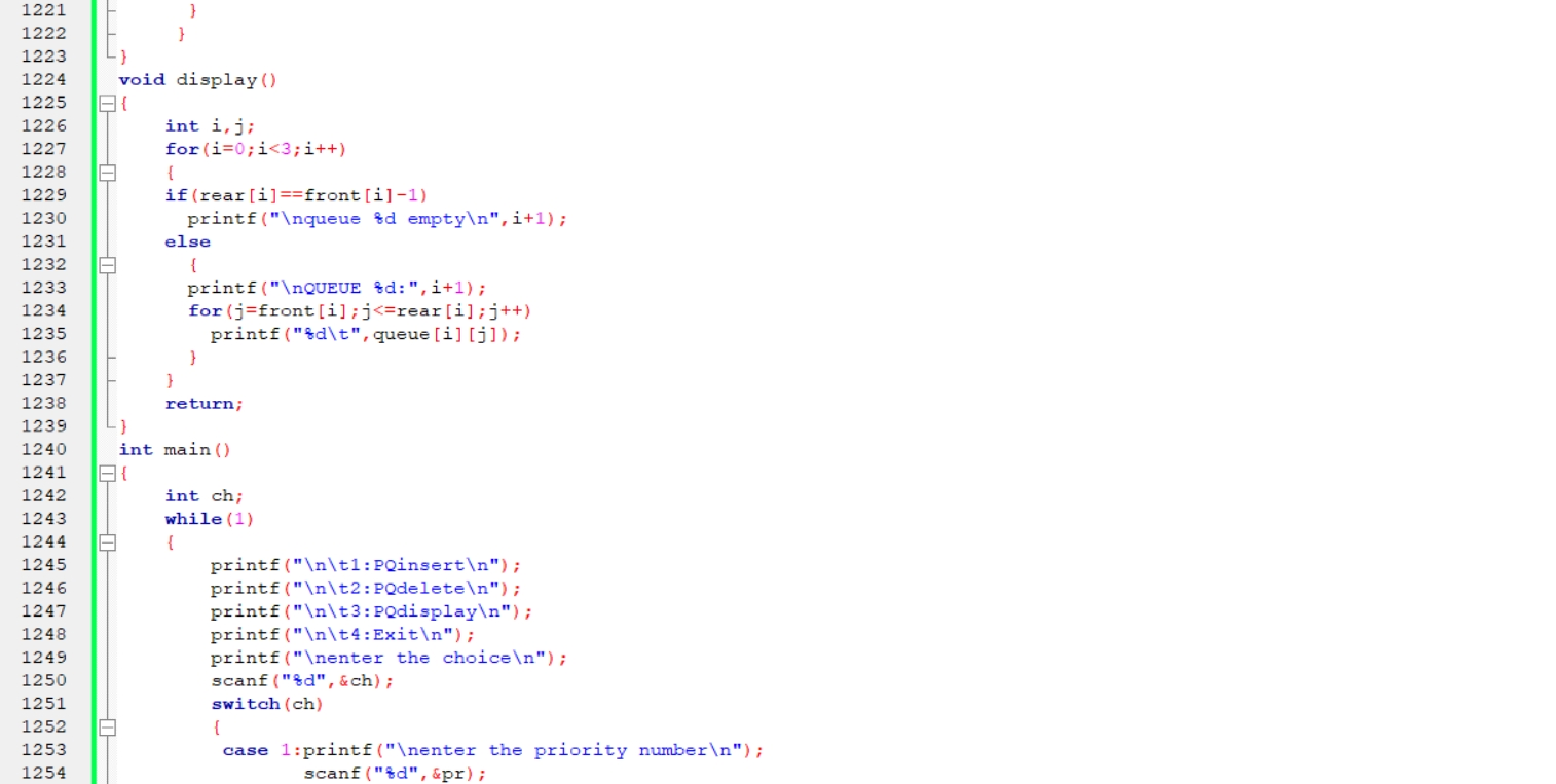
```
        }
    }while(choice!=5);
}
int main()
{   int choice;
    printf("1.Input restricted dequeue\n");
    printf("2.Output restricted dequeue\n");
    printf("3.Exit\n");
    printf("Enter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
     case 1 :input_que();
        break;
     case 2:output_que();
        break;
     case 3:exit(0);
     default:printf("Wrong choice\n\n");
    }
    return 0;
}
```


"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

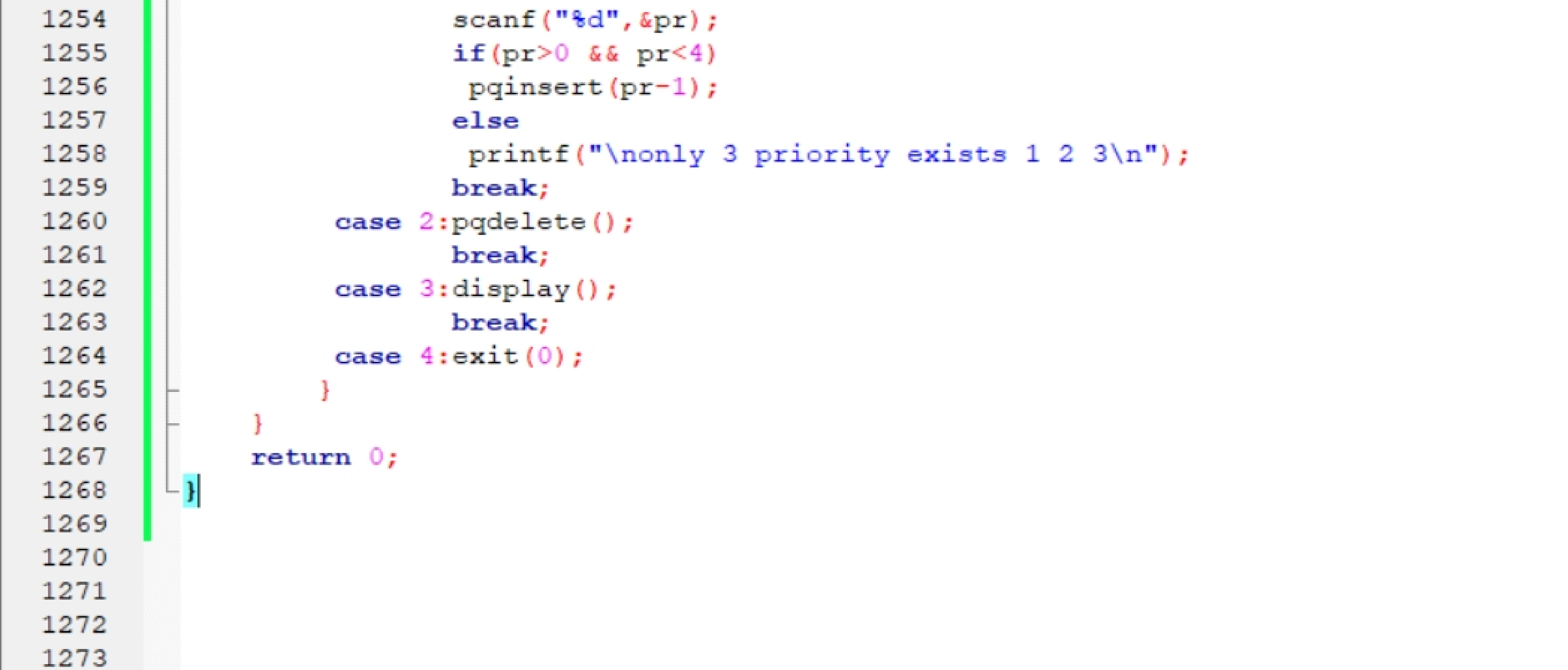
```
1.Input restricted dequeue
2.Output restricted dequeue
3.Exit
Enter your choice : 1
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 1
Enter the element : 23
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 1
Enter the element : 65
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 3
Element deleted from queue is : 65
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 4
Queue elements :
23
1.Insert at rear
2.Delete from front
3.Delete from rear
4.Display
5.Quit
Enter your choice : 5

Process returned 0 (0x0)   execution time : 18.298 s
Press any key to continue.
```



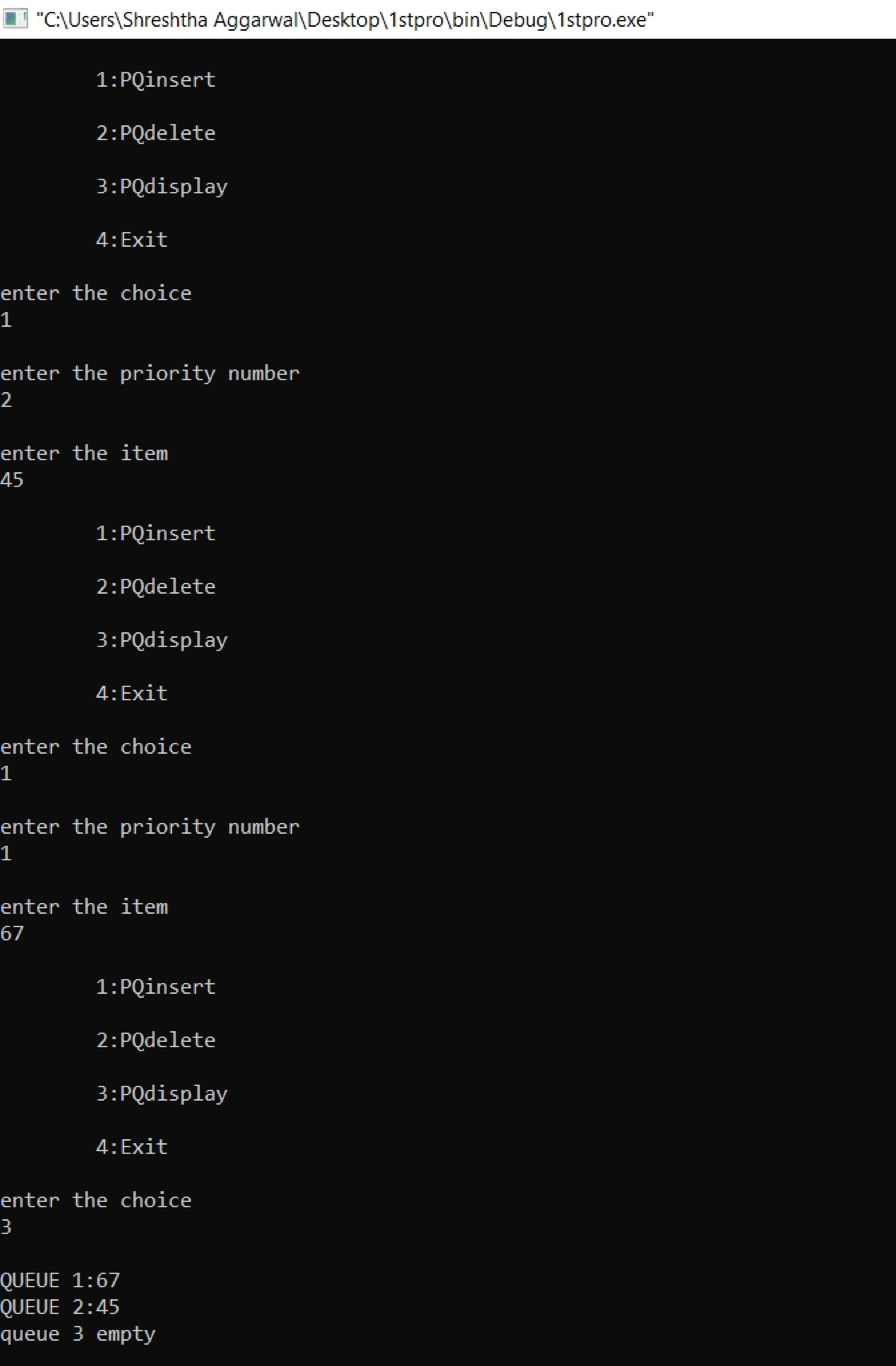
```
/*Multiple Priority Queue*/
#include<stdio.h>
#include<stdlib.h>
#define N 3
int queue[3][N];
int front[3]={0,0,0};
int rear[3]={-1,-1,-1};
int item,pr;
void pqinsert(int pr)
{
     if(rear[pr]==N-1)
        printf("\n Queue overflow\n");
     else
     {
         printf("\nenter the item\n");
         scanf("%d",&item);
         rear[pr]++;
         queue[pr][rear[pr]]=item;
     }
     return;
}
void pqdelete()
{
    int i;
    for(i=0;i<3;i++)
    {
      if(rear[i]==front[i]-1)
       printf("\nqueue %d empty\n",i+1);
      else
      {
       printf("\ndeleted item is %d of queue %d\n",queue[i][front[i]],i+1);
       front[i]++;
       return;
      }
```



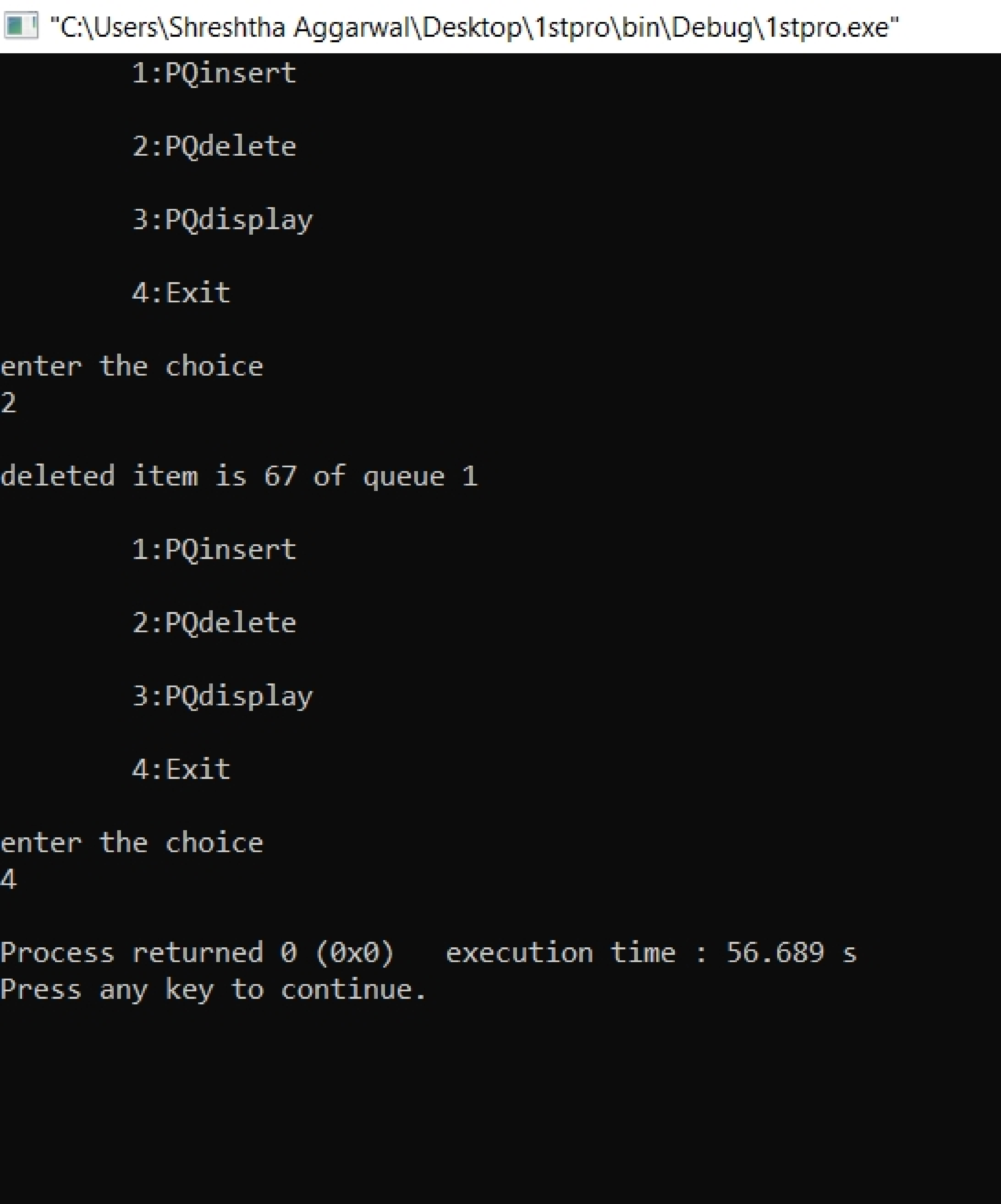
```c
      }
     }
}
void display()
{
    int i,j;
    for(i=0;i<3;i++)
    {
    if(rear[i]==front[i]-1)
      printf("\nqueue %d empty\n",i+1);
    else
      {
      printf("\nQUEUE %d:",i+1);
      for(j=front[i];j<=rear[i];j++)
        printf("%d\t",queue[i][j]);
      }
    }
    return;
}
int main()
{
    int ch;
    while(1)
    {
        printf("\n\t1:PQinsert\n");
        printf("\n\t2:PQdelete\n");
        printf("\n\t3:PQdisplay\n");
        printf("\n\t4:Exit\n");
        printf("\nenter the choice\n");
        scanf("%d",&ch);
        switch(ch)
        {
         case 1:printf("\nenter the priority number\n");
                scanf("%d",&pr);
```



```c
                scanf("%d",&pr);
                if(pr>0 && pr<4)
                 pqinsert(pr-1);
                else
                 printf("\nonly 3 priority exists 1 2 3\n");
                break;
         case 2:pqdelete();
                break;
         case 3:display();
                break;
         case 4:exit(0);
        }
    }
    return 0;
}
```


"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
        1:PQinsert

        2:PQdelete

        3:PQdisplay

        4:Exit

enter the choice
1

enter the priority number
2

enter the item
45

        1:PQinsert

        2:PQdelete

        3:PQdisplay

        4:Exit

enter the choice
1

enter the priority number
1

enter the item
67

        1:PQinsert

        2:PQdelete

        3:PQdisplay

        4:Exit

enter the choice
3

QUEUE 1:67
QUEUE 2:45
queue 3 empty
```


"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
        1:PQinsert

        2:PQdelete

        3:PQdisplay

        4:Exit

enter the choice
2

deleted item is 67 of queue 1

        1:PQinsert

        2:PQdelete

        3:PQdisplay

        4:Exit

enter the choice
4

Process returned 0 (0x0)   execution time : 56.689 s
Press any key to continue.
```