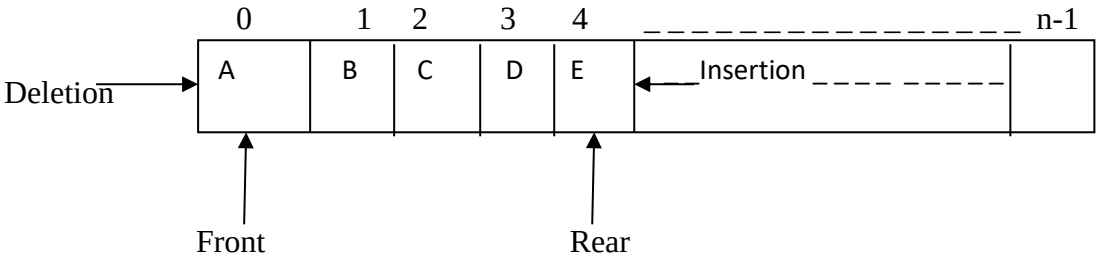
Practical no -8

Aim: write a program for implementation of a queue using one dimensional array.

Introduction

A *queue* is also a list of elements with insertions permitted at one end—called the rear, and deletions permitted from the other end—called the front. This means that the removal of elements from a queue is possible in the same order in which the insertion of elements is made into the queue. Thus, a queue data structure exhibits the *FIFO* (*first in first out*) property. insert and delete are the operations that are provided for insertion of elements into the queue and the removal of elements from the queue, respectively.



Algorithm:

For Insertion operation:

Steps:

Insert (array[],element)

1.[check for overflow]

If rear>=size

Printf ”Queue is overflow” and return.

2.[increment the pointer i.e rear by 1]

rear=rear+1

3.[perform insertion]

Array [rear] =element.

4. Exit

For Deletion operation:

Steps:

Delete (array [])

1.[check for underflow]

If front= -1

Print “Queue is underflow” and exit.

2. [Check for empty queue]

If front==rear

Front=0

Rear=0

Else

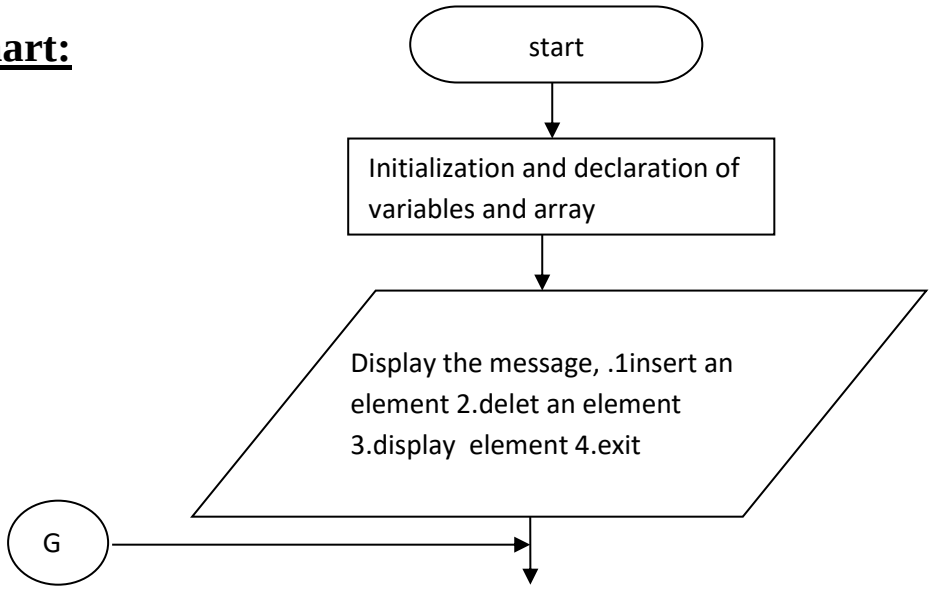
Front=front+1

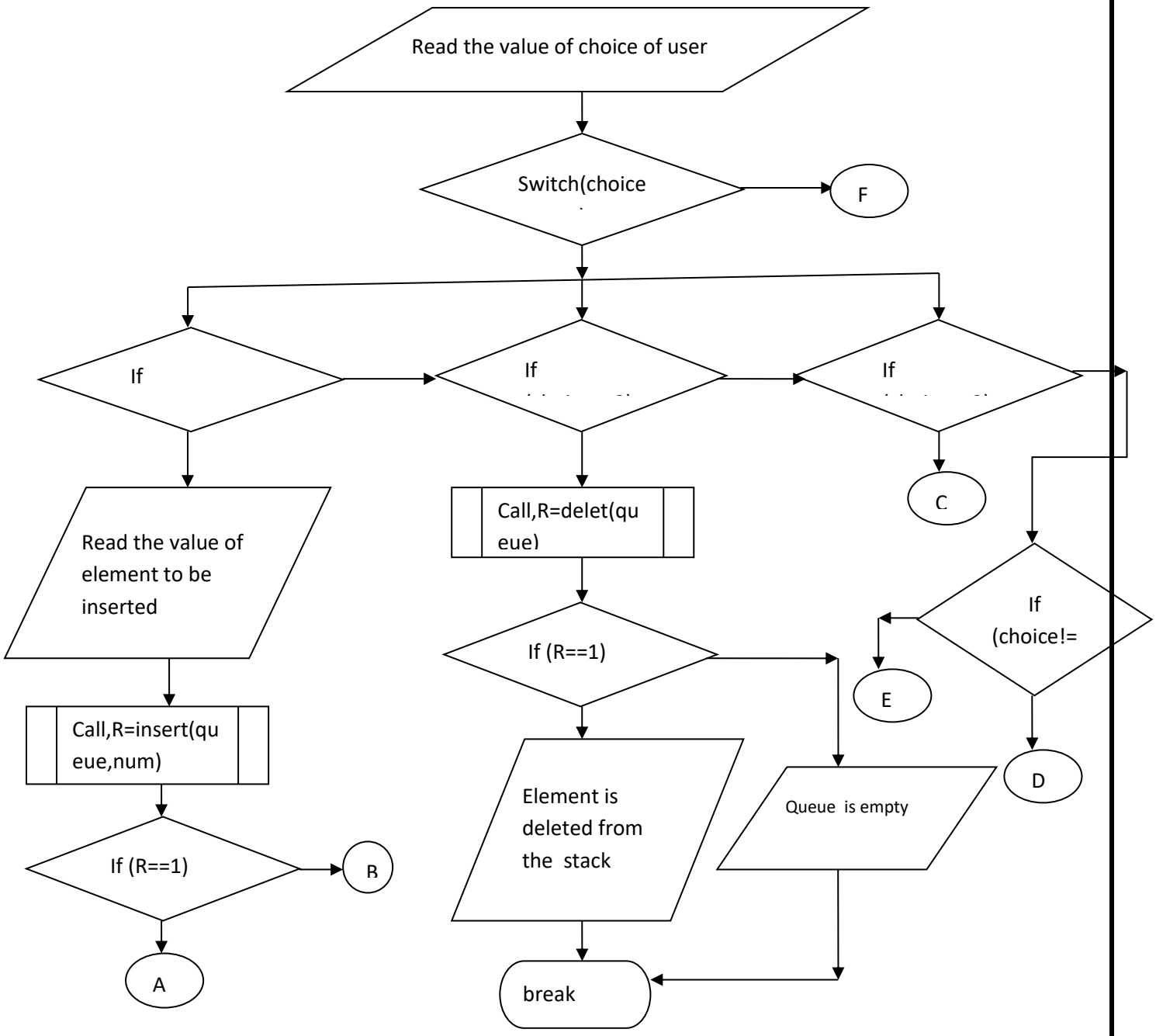
3. [Return the deleted element]

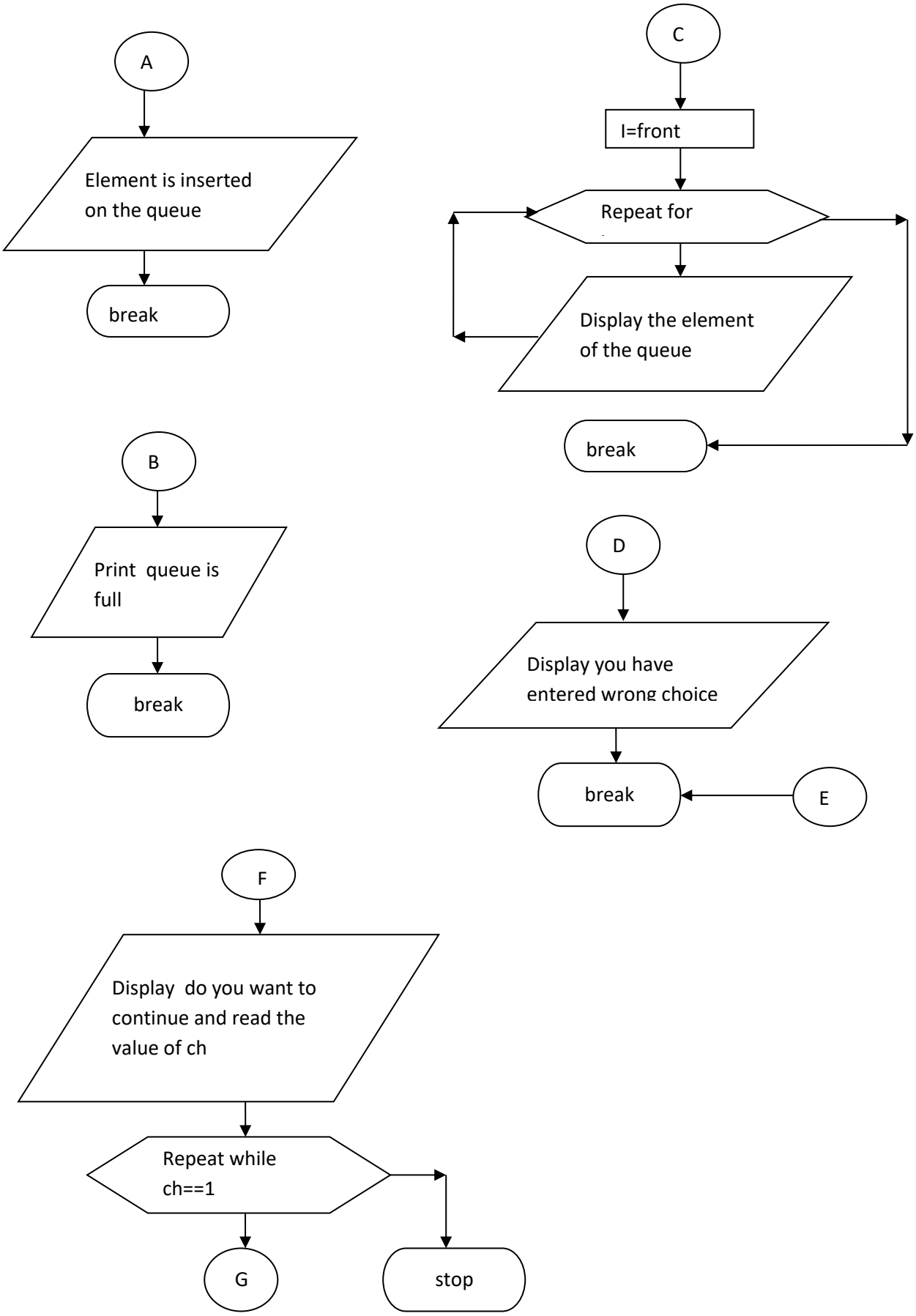
Return (element).

4. Exit

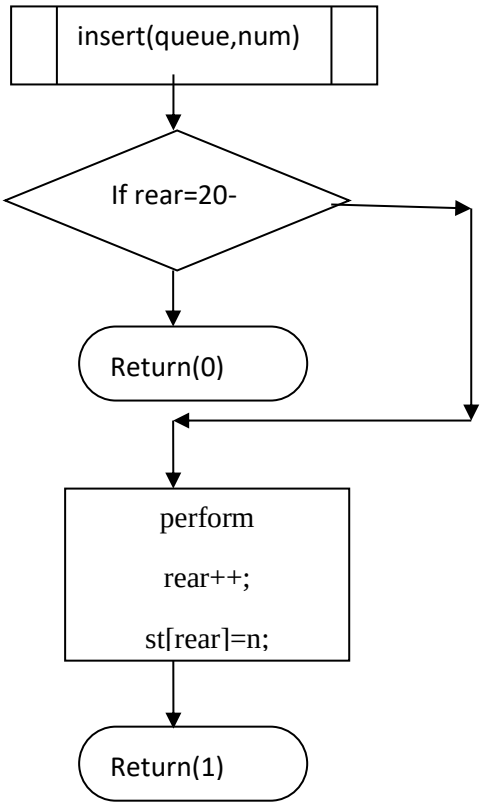
Flowchart:




Read the value of choice of user
Switch(choice
F
If
If
If
Read the value of element to be inserted
Call,R=delet(queue)
C
If (choice!=
Call,R=insert(queue,num)
If (R==1)
E
If (R==1)
B
Element is deleted from the stack
Queue is empty
D
A
break

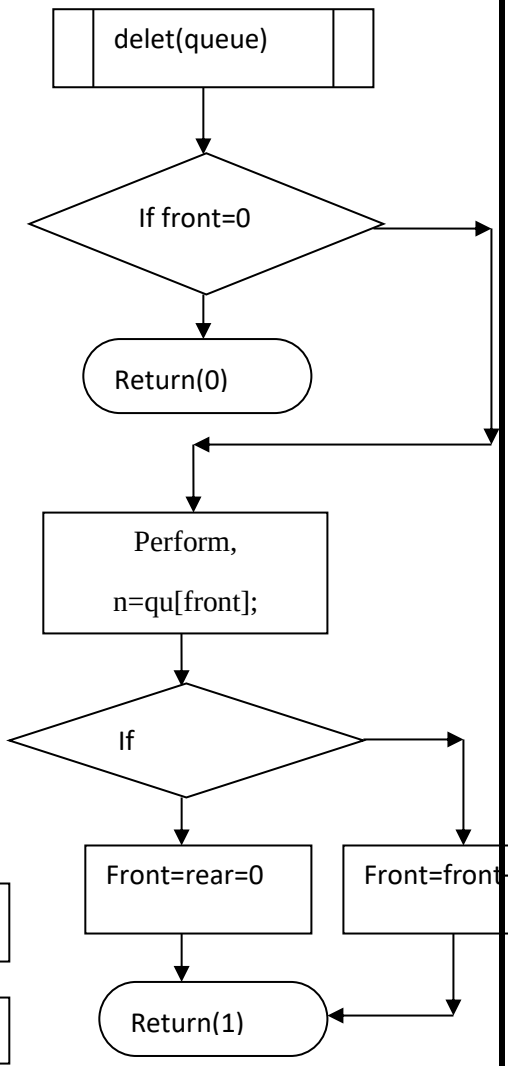

A
Element is inserted on the queue
break
C
I=front
Repeat for
Display the element of the queue
break
B
Print queue is full
break
D
Display you have entered wrong choice
break
E
F
Display do you want to continue and read the value of ch
Repeat while ch==1
G
stop

For insert function

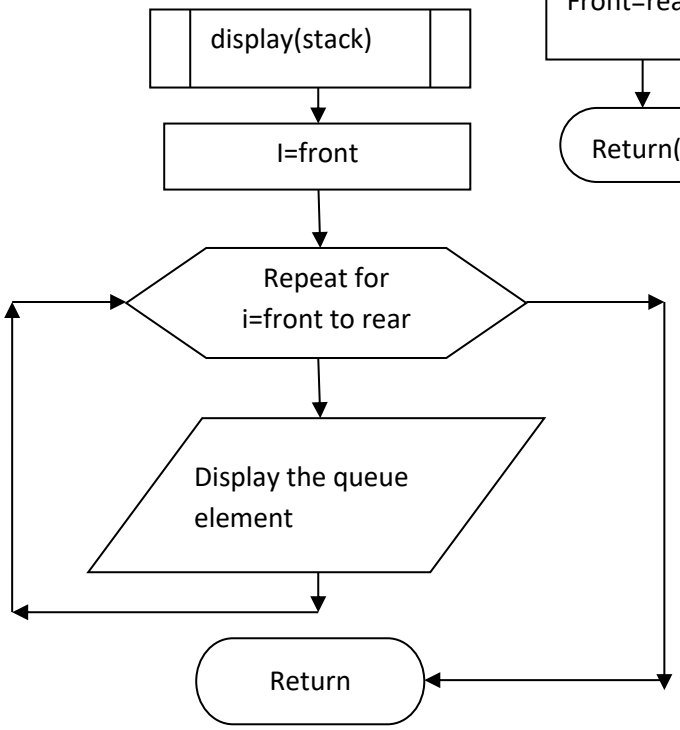


function

For delete



For display function:



Program:

```
#include<stdio.h>
#include<conio.h>
int front=-1;
int rear=-1;
void main()
{
        int queue[20],ch,num,r,choice,i;
        int insert(int[],int);
        int delet(int[]);
        void display(int[]);
        clrscr();
        printf("\n1.Insert an element \n2.Delet an element \n3.Display Queue element \n4.exit");
        do
        {       printf("\n Enter ur choice:");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                                printf("\nEnter any element:");
                                scanf("%d",&num);
                                r=insert(queue,num);
                                if(r==1)
                                printf("\nElement insert in to the queue.");
                                else
                                printf("\nQueue is full");
                                break;
                        case 2:
```

```
                    r=delet(queue);
                    if(r==1)
                    printf("\nElement is deleted from the queue.");
                    else
                    printf("\nQueue is empty.");
                    break;
            case 3:
                    display(queue);
                    break;
            default :
                    printf("\n\nU have entered wrong choice.");
        }

        printf("\n\nDo u want to continue y/n .");
        scanf("%d",&ch);
    }while(ch==1);
    getch();
}
int insert(int qu[],int n)
{
    if(rear==20-1)
    {
    return(0);
    }
    else
    {
        rear++;
        qu[rear]=n;
```

```
                if(front==-1)
                front=0;
                //int i=front+1;
                return(1);
        }

}
int delet(int qu[])
{
        if(front== -1)
        {
        return(0);
        }
        else
        {
             qu[front]= -1;
                if(front==rear)
            {
            front=0;
            rear=0;
            }
            else
            front=front+1;
            return(1);
        }
}
void display(int qu[])
{
```

```
        int i=front;
        while(i<=rear)
        {
        printf("%d\n",qu[i]);
        i++;
        }
}
```

Output:

1.Insert an element

2.Delet an element

3.Display Queue element

4.exit

Element insert in to the queue.

Do u want to continue y/n .1

Enter ur choice:1

Enter any element:20

Element insert in to the queue.

Do u want to continue y/n .1

Enter ur choice:1

Enter any element:30

Element insert in to the queue.

Do u want to continue y/n .1

Enter ur choice:1

Enter any element:40

Element insert in to the queue.

Do u want to continue y/n .1

Enter ur choice:3

10

20

30

40

Do u want to continue y/n .1

Enter ur choice:2

Element is deleted from the queue.

Do u want to continue y/n .1

Enter ur choice:3

20

30

40

Do u want to continue y/n .0