**#4. Queue**

**Roll Number: cb.en.p2ebs22006**

**Date of Submission: 21-11-2022**

**Aim:**

To perform following Queue operations using C Programming:

1. Queue Creation
2. Add an element
3. Remove an element
4. Check the status of the queue full/empty
5. View the entire queue after each of the above operation

**Tools Required:**

Text editor with C Compiler.

**Experiment:**

Code:

#include <stdio.h>

#include <stdlib.h>

typedef struct Queue{

unsigned int size;

int front;

int rear;

int \*array;

}Queue1;

Queue1\* create(unsigned int size){

Queue1 \*q;

q= (struct Queue\*)malloc(sizeof(struct Queue));

q->array=(int\*)malloc(size\*sizeof(int));

q->size=size-1;

q->front=-1;

q->rear=-1;

return q;

}

int isFull(Queue1 \*q){

if(q->rear==q->size){

printf("Queue is Full");

}

return q->rear!=q->size;

}

int isEmpty(Queue1 \*q){

if(q->front==-1){

printf("Queue is Empty");

}

return q->front==-1;

}

void enqueue(Queue1 \*q){

int queueElement;

if(isFull(q)){

if(q->front==-1)

q->front=0;

printf("Please enter the number which needs to be saved in the queue:");

scanf("%d",&queueElement);

q->array[++q->rear]=queueElement;

}

}

void dequeue(Queue1 \*q){

if(q->front==q->rear){

q->array[q->front++]=NULL;

q->front=q->rear=-1;

}

if(!isEmpty(q)){

q->array[q->front++]=NULL;

}

}

void peek(Queue1 \*q){

printf("%d",q->array[q->front]);

}

void view(Queue1 \*q){

int i,value;

value=(q->front!=-1)?q->front:0;

for(i=value;i<=q->size;i++){

printf("\n---------\n");

printf("|\t");

if(!q->array[i])

printf("\t");

if(q->array[i])

printf("%d\t",q->array[i]);

printf("|\n");

printf("---------\n");

}

}

int main() {

Queue1 \*q;

unsigned int size, userInput;

printf("Enter the number of queue elements:");

scanf("%d",&size);

q=create(size);

while(1){

printf("\n1-Enqueue\n2-Dequeue\n3-isFull()\n4-isEmpty()\n5-peek()\n6-view()\n7-exit\n");

scanf("%d",&userInput);

switch(userInput){

case 1:enqueue(q);

break;

case 2:dequeue(q);

break;

case 3:isFull(q);

break;

case 4:isEmpty(q);

break;

case 5:peek(q);

break;

case 6:view(q);

break;

case 7: exit(1);

break;

default: printf("Enter a valid number");

}

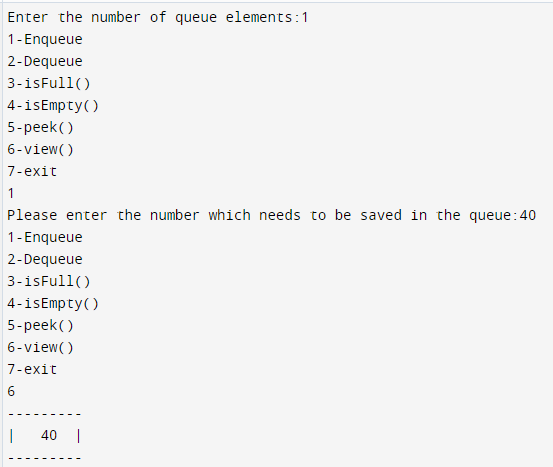
}

return 0;

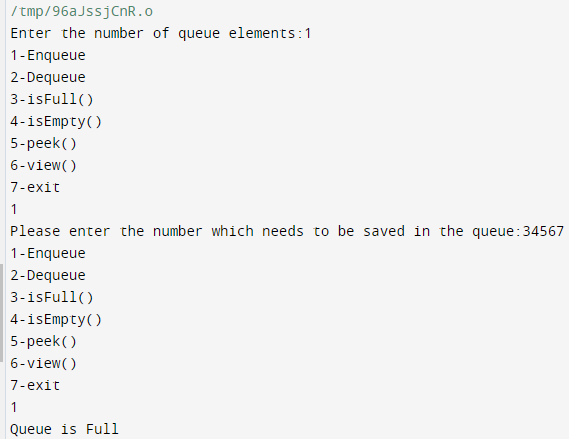
}

Result:

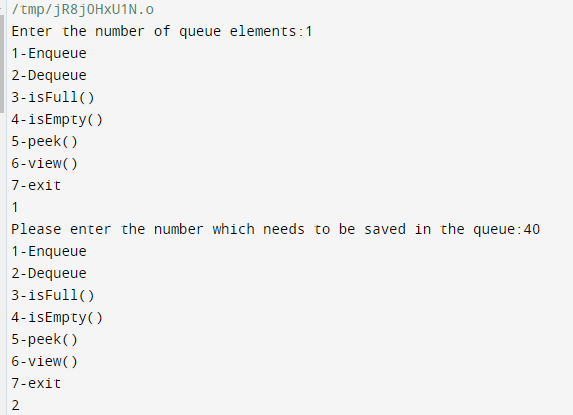
1. Queue Creation

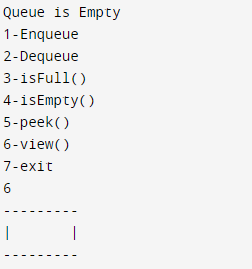


1. Add an element

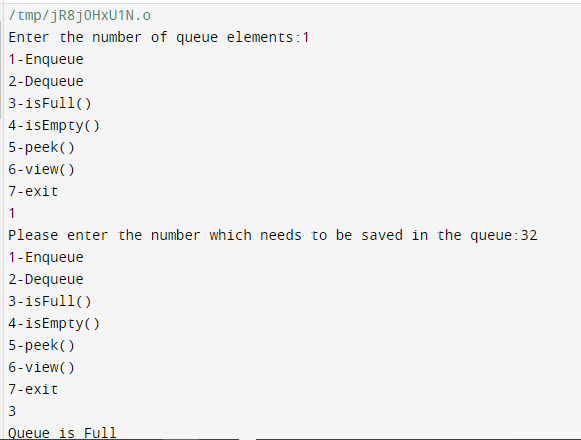


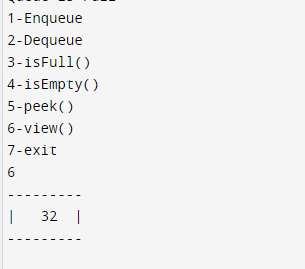
1. **Remove an element**

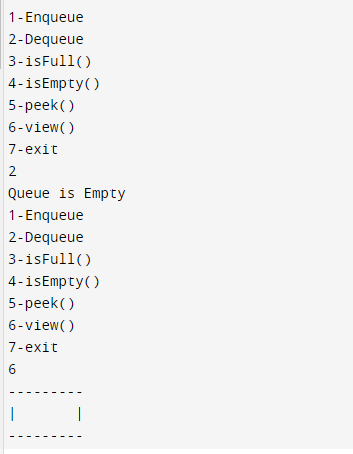
****

****

1. Check the status of the queue full or empty

****

****

****

**Inference and Result:**

Implementation of Queue along with functions such as creation, enqueue, dequeue, checking the status of the queue, viewing the elements has been done and the corresponding results has been obtained.