**#4. Queue**

**Roll Number:CB.EN.P2EBS22002**

**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;

}Q;

Q\* create(unsigned int size){

Q \*queue;

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

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

queue->size=size-1;

queue->front=-1;

queue->rear=-1;

return queue;

}

int isFull(Q \*queue){

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

printf("Queue is Full");

}

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

}

int isEmpty(Q \*queue){

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

printf("Queue is Empty");

}

return queue->front==-1;

}

void enqueue(Q \*queue){

int queueElement;

if(isFull(queue)){

if(queue->front==-1)

queue->front=0;

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

scanf("%d",&queueElement);

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

}

}

void dequeue(Q \*queue){

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

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

queue->front=queue->rear=-1;

}

if(!isEmpty(queue)){

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

}

}

void peek(Q \*queue){

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

}

void view(Q \*queue){

int i,value;

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

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

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

printf("|\t");

if(!queue->array[i])

printf("\t");

if(queue->array[i])

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

printf("|\n");

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

}

}

int main() {

Q \*queue;

unsigned int size, userInput;

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

scanf("%d",&size);

queue=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(queue);

break;

case 2:dequeue(queue);

break;

case 3:isFull(queue);

break;

case 4:isEmpty(queue);

break;

case 5:peek(queue);

break;

case 6:view(queue);

break;

case 7: exit(1);

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

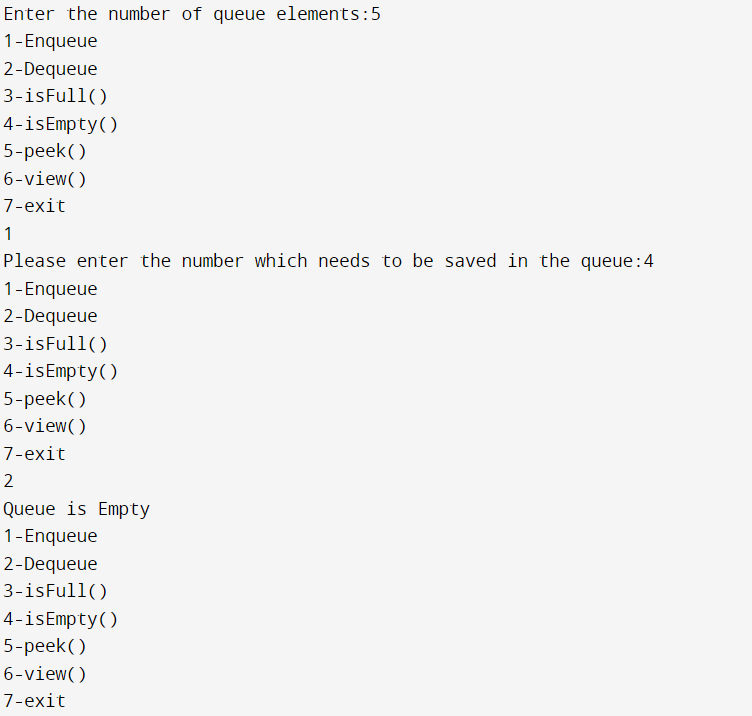
}

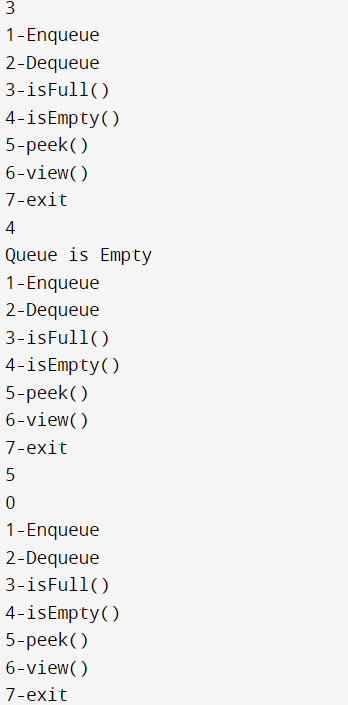
}

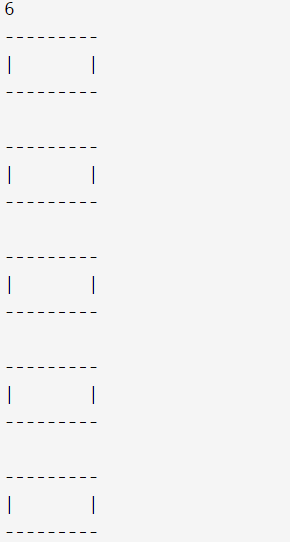
return 0;

}

Result







**Inference and Result:**