**#4. Queue**

**Roll Number:cb.en.p2p2ebs22005**

**Date of Submission:21-11-22**

**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 {

int size;

int rear;

int front;

int \*array;

}Q1;

Q1\* create(unsigned int size)

{

Q1 \*memory;

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

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

memory->size=size-1;

memory->front=-1;

memory->rear=-1;

return memory;

}

int isfull(Q1 \*q1){

if(q1->rear==q1->size){

return printf("Queue is Full");

}

else

{

return 1;

}

}

int isempty(Q1 \*q1)

{

if(q1->front==-1&&q1->rear==-1)

{

printf("The queue is empty");

}

else

{

printf("The queue is not empty");

}

}

void enqueue(Q1 \*q1){

int number;

if(isfull(q1)==1){

if(q1->front==-1)

{

q1->front=0;

}

printf("enter the number to be added to the queue:");

scanf("%d",&number);

q1->array[++q1->rear]=number;

}

}

void dequeue(Q1 \*q1)

{

if(!q1->array[q1->rear])

{

printf("The queue is already empty");

q1->front=-1;

q1->rear=-1;

}

else

{

q1->array[q1->front]=NULL;

q1->front++;

}

}

void peek(Q1 \*q1)

{

printf("The latest entry element is : %d",q1->array[q1->rear]);

}

void view(Q1 \*q1)

{

int i,value;

value=(q1->front!=-1)?q1->front:0;

for(i=value;i<=q1->size;i++)

{

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

printf("|\t");

if(!q1->array[i])

printf("\t");

if(q1->array[i])

printf("%d\t",q1->array[i]);

printf("|\n");

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

}

}

int main()

{

int n;

Q1 \*q1;

int Input;

printf("enter the size of the queue");

scanf("%d",&n);

q1=create(n);

while(1)

{

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

scanf("%d",&Input);

switch(Input){

case 1:enqueue(q1);

break;

case 2:dequeue(q1);

break;

case 3:isfull(q1);

break;

case 4:isempty(q1);

break;

case 5:peek(q1);

break;

case 6:view(q1);

break;

case 7: exit(0);

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

}

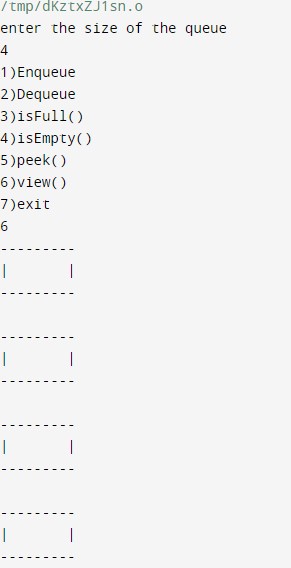
}

return 0;

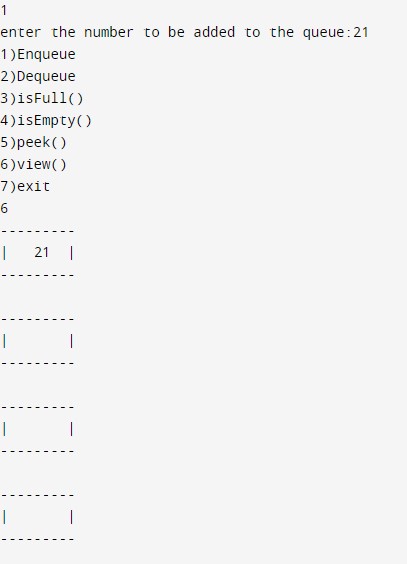
}

Result:

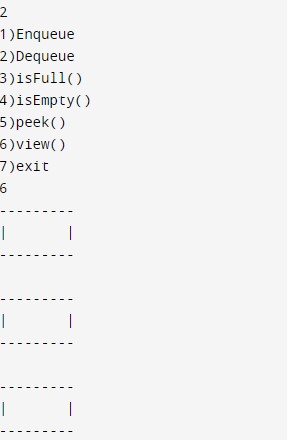
1. Queue Creation



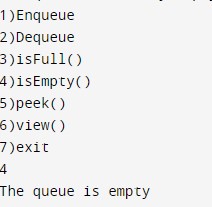
1. Add an element



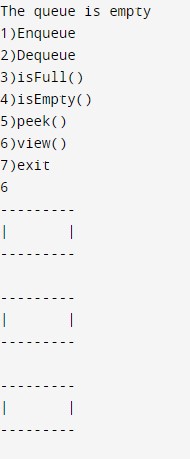
1. Remove an element



1. Check the status of the queue full/empty



1. View the entire queue after each of the above operation



**Inference and Result:**

Implementation of queue has been performed , functions such as enqueuer ,dequeuer, view peek and other functions have also been implemented.