DEPARTMENT OF INFORMATION TECHNOLOGY

Experiment No: 2

Aim: To implement Queue using Array.

Theory: A queue is a linear data structure that follows the First-In-First-Out (FIFO) principle. This means that the first element added to the queue will be the first one to be removed.

Key Operations

1. Enqueue: Add an element to the back of the queue.
2. Dequeue: Remove an element from the front of the queue.
3. Front: Get the element at the front of the queue without removing it.
4. IsEmpty: Check if the queue is empty.
5. IsFull: Check if the queue is full (if using a fixed-size array).

Theory:

Implementation Using Array

1. Array Representation: Use a fixed-size array to store elements of the queue. Keep track of the front and rear indices.
2. Initialization:
   - `front` and `rear` indices are initialized to `-1` to indicate an empty queue.
   - `size` is the maximum capacity of the queue.
3. Enqueue Operation:
   - Check if the queue is full.
   - If the queue is empty, set both `front` and `rear` to `0`.
   - Otherwise, increment the `rear` index in a circular manner (i.e., `(rear + 1)%size`).
   - Add the new element at the `rear` index.
4. Dequeue Operation:
   - Check if the queue is empty.
   - Retrieve the element at the `front` index.
   - If the queue becomes empty after the operation, reset `front` and `rear` to `-1`.
   - Otherwise, increment the `front` index in a circular manner.
5. Front Operation:
   - Simply return the element at the `front` index.

6. IsEmpty Operation:
   - The queue is empty if `front` is `-1`.
7. IsFull Operation:
   - The queue is full if `(rear + 1) % size == front`.

Code:

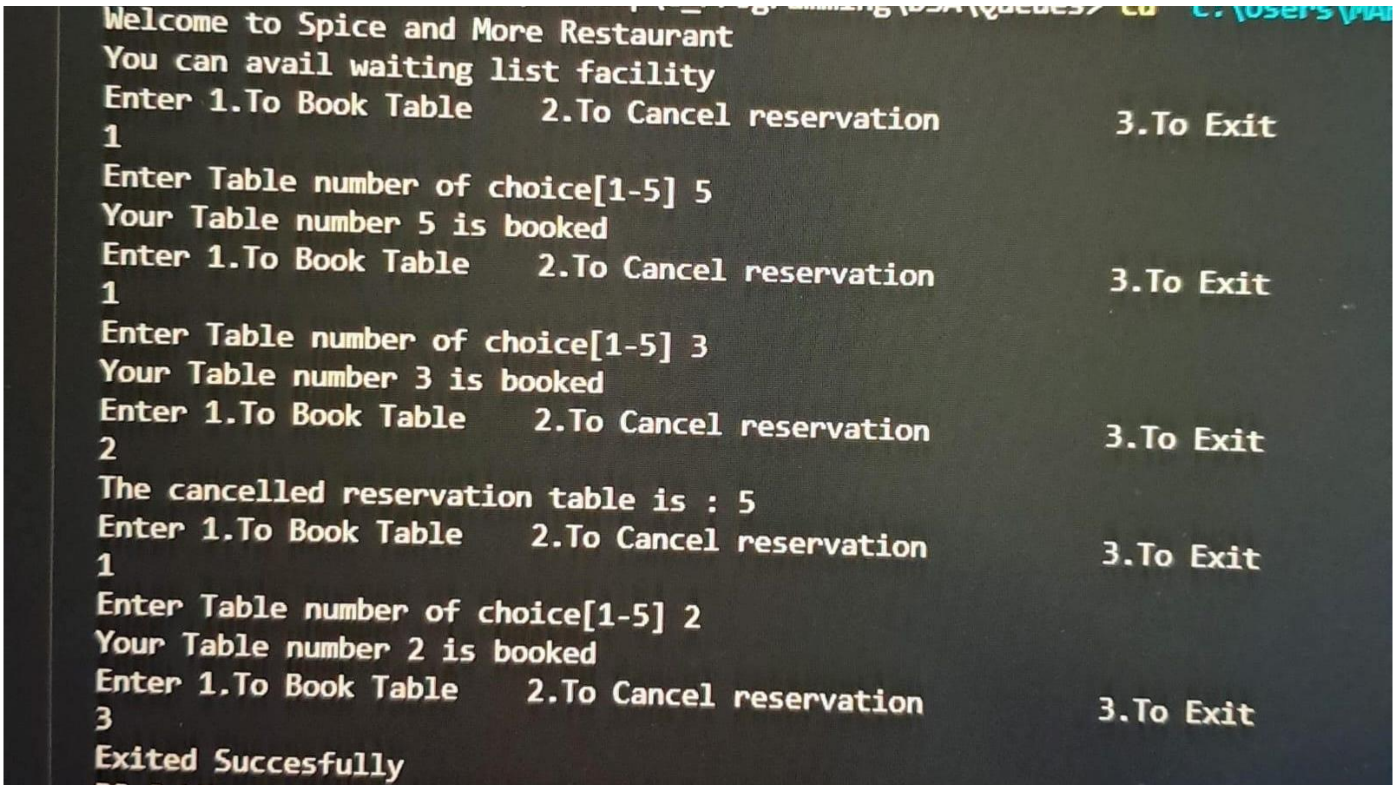
```
#include<stdio.h>
int Q [100], front=-1, rear=-1, x, i, n=5, choice;
void Display ();
void Insert ();
void Delete ();
void main () {
printf ("Welcome to Spice and More Restaurant\n");
printf ("You can avail waiting list facility\n");

do {

printf ("Enter 1. To Book Table\t 2. To Cancel reservation\t 3. To Exit\n");
scanf ("%d", &choice);
        switch(choice)
        {
         case 1: Insert ();
                 break;
        case 2: Delete ();
                break;
        case 3: printf ("Exited Successfully\n");
                break;
        }
  }
  while (choice! =3);
}
void Insert () {
if(rear>=n-1) {
printf ("Sorry, No Reservation available");
}
else {
printf ("Enter Table number of choice [1-5] ");
scanf ("%d", &x);
rear++;
Q[rear]=x;
if(front==-1)
{
front=0;
        }
        printf ("Your Table number %d is booked\n", x);
        }
}
void Delete () {
if (front==-1)
        {
```

```
        printf ("No Reservations to Cancel\n");
        }
        else {
        printf ("The cancelled reservation table is: %d\n”,Q[front]);
        }
        if(front==rear)
        front=rear=-1;
        else
        front++;
}
```

Output:



```
Welcome to Spice and More Restaurant
You can avail waiting list facility
Enter 1.To Book Table    2.To Cancel reservation           3.To Exit
1
Enter Table number of choice[1-5] 5
Your Table number 5 is booked
Enter 1.To Book Table    2.To Cancel reservation           3.To Exit
1
Enter Table number of choice[1-5] 3
Your Table number 3 is booked
Enter 1.To Book Table    2.To Cancel reservation           3.To Exit
2
The cancelled reservation table is : 5
Enter 1.To Book Table    2.To Cancel reservation           3.To Exit
1
Enter Table number of choice[1-5] 2
Your Table number 2 is booked
Enter 1.To Book Table    2.To Cancel reservation           3.To Exit
3
Exited Succesfully
```

Conclusion: The Queues Program has provided insights into Data Structures and Algorithm Concepts. Through the program a better understanding of Queue operations such as enqueue and dequeue is gained. The menu driven program has helped in understanding practical applications in real world scenarios

Submitted Details -

Name of Student: Awani Goyal

Roll No.: 31

Date of Performance: 08/07/2024