Name:-Aakash Bhatt
Sap-id:- 500124633

Data Structure

Lab-9

Submitted by:

Aakash Bhatt

500124633

(2023-2024)

Submitted to:

Pankaj Sir

Name:-Aakash Bhatt
Sap-id:- 500124633

1. Create Queue
2. Perform Enqueue and Deque operations on Queue.
3. Traverse the queue and print its element.
4. Print underflow and overflow when desired conditions are not met.

Reverse the elements of Queue using recursion

```c
#include <stdio.h>
#define N 5

int front = -1, rear = -1;
int queue[N];

void enqueue(int x)
{
    if (rear == N - 1)
    {
        printf("Overflow\n");
    }
    else if (front == -1 && rear == -1)
    {
        front = rear = 0;
        queue[rear] = x;
    }
    else
    {
        rear++;
        queue[rear] = x;
    }
}

void dequeue()
{
    if (front == -1 && rear == -1)
    {
        printf("Underflow\n");
    }
    else if (front == rear)
    {
        printf("The dequeue element is %d\n", queue[front]);
        front = rear = -1;
    }
    else
    {
        printf("The dequeue element is %d\n", queue[front]);
        front++;
    }
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
}

void peek()
{
    if (front == -1 && rear == -1)
    {
        printf("Queue is empty\n");
    }
    else
    {
        printf("%d\n", queue[front]);
    }
}

// Recursive function to reverse the queue
void reverseQueue(int i)
{
    if (i <= rear)
    {
        reverseQueue(i + 1);
        printf("%d ", queue[i]);
    }
}

void display()
{
    if (front == -1 && rear == -1)
    {
        printf("Queue is empty\n");
    }
    else
    {
        for (int i = front; i <= rear; i++)
        {
            printf("%d ", queue[i]);
        }
        printf("\n");
    }
}

int main()
{
    int choice, element;

    do
    {
        printf("Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to
display, 5 to reverse display, and 0 to exit: ");
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter the element to enqueue: ");
            scanf("%d", &element);
            enqueue(element);
            break;

        case 2:
            dequeue();
            break;

        case 3:
            peek();
            break;

        case 4:
            display();
            break;

        case 5:
            printf("Reversed queue elements: ");
            reverseQueue(front);
            printf("\n");
            break;

        case 0:
            printf("Exiting the program\n");
            break;

        default:
            printf("Invalid choice\n");
        }
    } while (choice != 0);

    return 0;
}
```

Name:-Aakash Bhatt
Sap-id:- 500124633

```
PS D:\MCA\MCA-DSA\LAB-9> gcc .\Question1.c
PS D:\MCA\MCA-DSA\LAB-9> .\a.exe
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 2
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 4
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 6
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 1
Enter the element to enqueue: 8
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 4
2 4 6 8
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 5
Reversed queue elements: 8 6 4 2
Press 1 to enqueue, 2 to dequeue, 3 to peek, 4 to display, 5 to reverse display,
 and 0 to exit: 0
Exiting the program
```