

**Lab No: 4 Date: 2081/**

**Title: Write a Menu based program to show the basic operation of Linear Queue.**

A Queue Data Structure is a fundamental concept in computer science used for storing and managing data in a specific order. It follows the principle of "First in, First out" (FIFO), where the first element added to the queue is the first one to be removed. Queues are commonly used in various algorithms and applications for their simplicity and efficiency in managing data flow.

Basic Operations on Queue:

Some of the basic operations for Queue in Data Structure are:

* enqueue() – Insertion of elements to the queue.
* dequeue() – Removal of elements from the queue.
* peek() or front()- Acquires the data element available at the front node of the queue without deleting it.
* rear() – This operation returns the element at the rear end without removing it.
* isFull() – Validates if the queue is full.
* isEmpty() – Checks if the queue is empty.
* size(): This operation returns the size of the queue i.e. the total number of elements it contains.

**IDE: Visual Studio Code**

**Language: C**

**Source code:**

#include <stdio.h>

#include <stdlib.h>

#define MAX 5 // array size

int queue[MAX];

int front = -1;

int rear = -1;

void enqueue(int x)

{

    if (rear == MAX - 1)

    {

        printf("Overflow\n");

    }

    else if (front == -1 && rear == -1)

    {

        front = rear = 0;

        queue[rear] = x;

        printf("%d is enqueue\n", x);

    }

    else

    {

        rear++;

        queue[rear] = x;

        printf("%d is enqueue\n", x);

    }

}

void dequeue()

{

    if (front == -1 && rear == -1)

    {

        printf("Empty\n");

    }

    else if (front == rear)

    {

        printf("%d is dequeue\n", queue[front]);

        front = rear = -1;

    }

    else

    {

        printf("%d is dequeue\n", queue[front]);

        front++;

    }

}

void display()

{

    int i;

    if (front == -1 && rear == -1)

    {

        printf("Empty\n");

    }

    else

    {

        for (i = front; i < rear + 1; i++)

        {

            printf("%d,", queue[i]);

        }

    }

}

int main()

{

    int choose, n,item;

    while (1)

    {

        printf("\n\n1.For Enqueue\n");

        printf("2. For Dequeue\n");

        printf("3. For display\n");

        printf("4.For exit\n");

        printf("Choose number(1-4):- ");

        scanf("%d", &choose);

        switch (choose)

        {

        case 1:

            printf("Enter a number to enqueue:-");

            scanf("%d", &item);

            enqueue(item);

            break;

        case 2:

            dequeue();

            break;

        case 3:

            display();

            break;

        case 4:

            printf("\nProgram END...\n");

            exit(0);

        default:

            printf("Wrong Entry!!!\n Please choose (1-4) only\n");

            break;

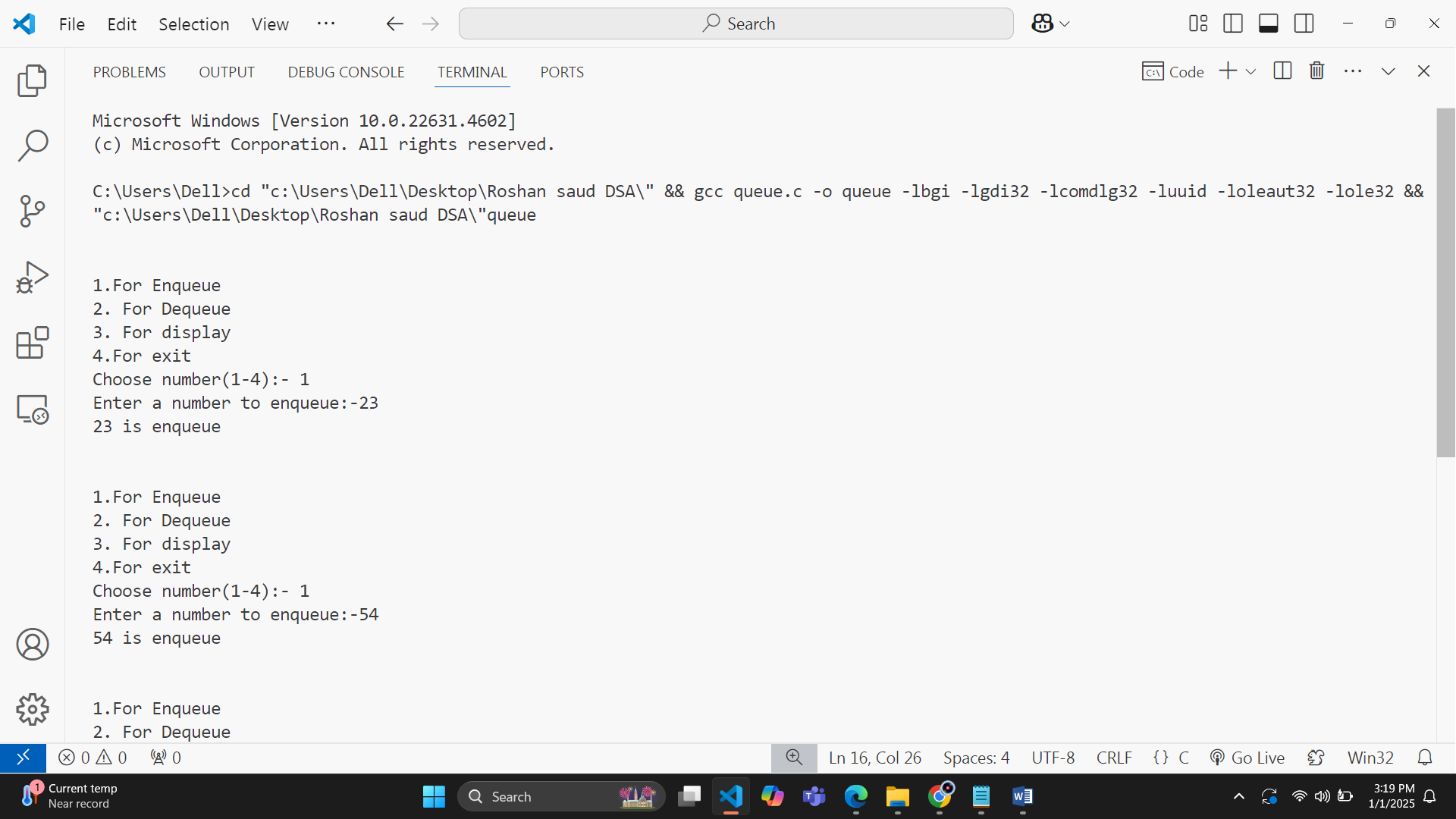
        }

    }

    return 0;

}

**Output:**

****

