**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Course Name and Code:** Data Structures Lab **(**ITL302)

**Semester:** III (SYIT)

**Academic Year:** 2024-25 (Odd Semester)

**Experiment No. 06**

**Aim:** **Implementation of Binary Tree and its Traversal for real-world application.**

**Code:**

#include <stdio.h>

#include <stdlib.h>

struct Queue

{

int N;

int Q[100];

int Front;

int Rear;

};

void Enqueue(struct Queue \* Qptr, int ele);

int Dequeue(struct Queue\* Qptr);

void Display(struct Queue \* Qptr);

int main()

{

struct Queue Qu;

printf("Enter size of Queue: ");

scanf("%d", &(Qu.N));

Qu.Front = -1, Qu.Rear = -1;

int choice, temp;

while(1)

{

printf("\nChoose an option for the Queue: \n");

printf("1. Enqueue\n2. Dequeue\n3. Display\n4. Exit\n");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter element to be Enqueueed: ");

scanf("%d", &temp);

Enqueue(&Qu, temp);

break;

case 2:

printf("Dequeueped Element: %d", Dequeue(&Qu));

break;

case 3:

printf("Queue: ");

Display(&Qu);

break;

default:

return 0;

}

}

return 0;

}

void Enqueue(struct Queue \* Qptr, int ele)

{

if(Qptr->Rear >= Qptr->N - 1)

{

printf("Queue Overflow Error!");

exit(-1);

}

else if(Qptr->Front == -1)

Qptr->Front += 1;

Qptr->Rear += 1;

Qptr->Q[Qptr->Rear] = ele;

}

int Dequeue(struct Queue \* Qptr)

{

if(Qptr->Front < 0 || Qptr->Front > Qptr->Rear)

{

printf("Queue Underflow Error!");

exit(-1);

}

int ele = Qptr->Q[Qptr->Front];

if(Qptr->Front == Qptr->Rear)

{

Qptr->Front = -1;

Qptr->Rear = -1;

}

else

Qptr->Front += 1;

return ele;

}

void Display(struct Queue \* Qptr)

{

printf("[");

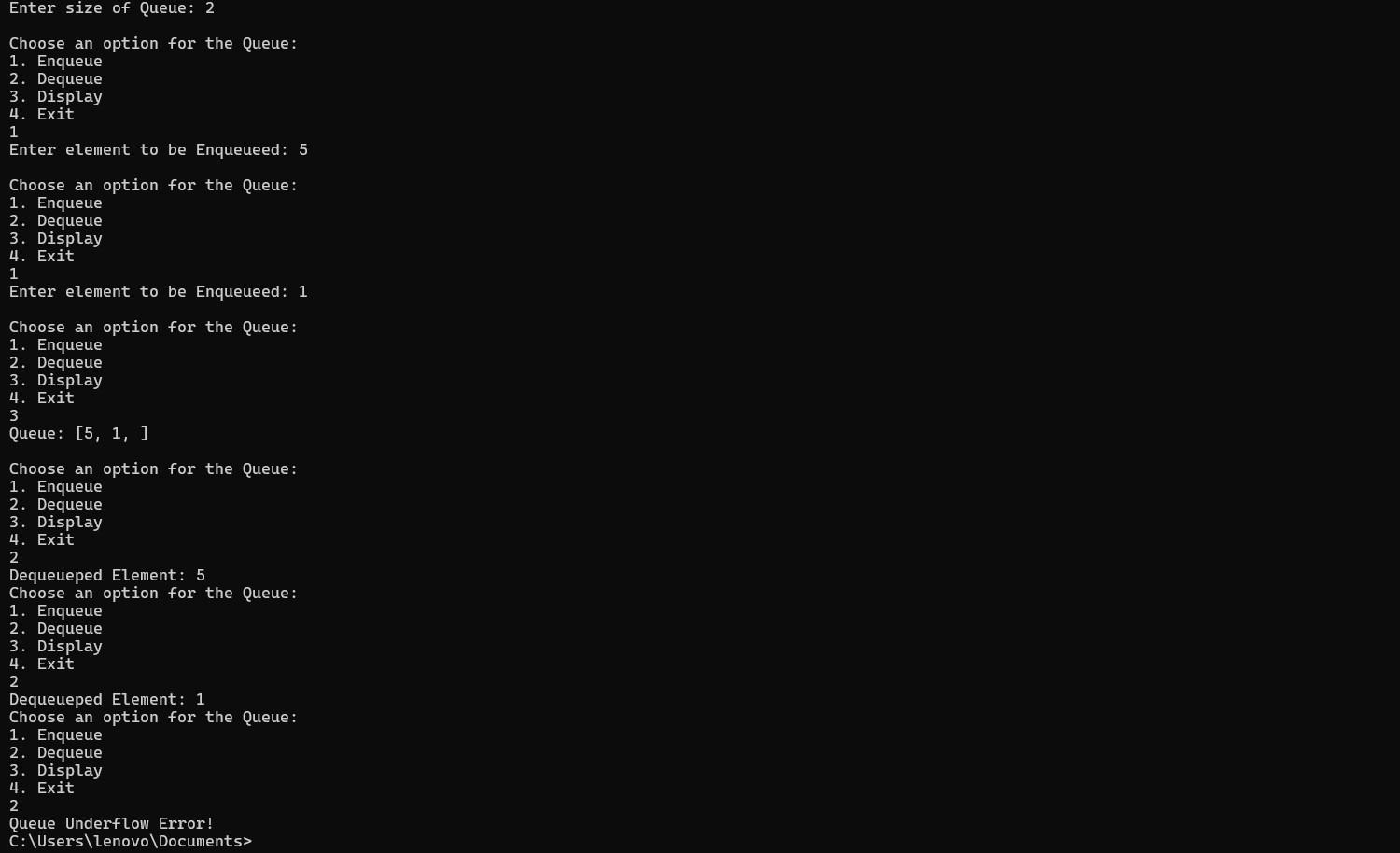
for(int i = Qptr->Front; i <= Qptr->Rear && Qptr->Rear != -1; i++)

printf("%d, ", Qptr->Q[i]);

printf("]\n");

}

**Output:**

****

**Submitter Details:-**

**Name:** Faizan Dodiya

**Roll No:** 24

**Div/Batch :** A/ S-2