# Green University of Bangladesh

**Department of Computer Science and Engineering (CSE)**
Faculty of Sciences and Engineering
Semester: Summer 2022, B.Sc. in CSE (DAY)

**LAB REPORT NO # 04**

**Course Title: Data Structure Lab**
**Course Code: CSE 106** **Section: CSE 213 - DA (PC)**

**Lab Experiment Name(s):**

- Circular queue.
- Implement a program for converting an infix expression to postfix expression using stack.

**Student Details**

| Name | ID |
|---|---|
| Md. Shahidul Islam Prodhan | 213902017 |

**Lab Date:** 23 July 2022

**Submission Date:** 07 August 2022

**Course Teacher's Name:** Ms Farhana Akter Sunny, Senior Lecturer.

**[For Teacher's use only: Don't write anything inside this box]**

**Lab Report Status**

| **Marks:** | **Signature:** |
|---|---|
| **Comments:** | **Date:** |

## 1. TITLE OF THE LAB EXPERIMENT

Write a C program to add ,delete ,display a circular queue.

## 2. OBJECTIVES

Implement Circular Queue

## 3. PROCEDURE/ ANALYSIS / DESIGN

***Problem 1: circular queue***

| STEPS | Algorithm to insert an element in a circular queue |
|---|---|
| 1 | IF (REAR+1)%MAX = FRONT<br>Write " OVERFLOW "<br>Goto step 4<br>[End OF IF] |
| 2 | IF FRONT = -1 and REAR = -1<br>SET FRONT = REAR = 0<br>ELSE IF REAR = MAX - 1 and<br>FRONT ! = 0<br>SET REAR = 0<br>ELSE<br>SET REAR = (REAR + 1) % MAX<br>[END OF IF] |
| 3 | SET QUEUE[REAR] = VAL |
| 4 | EXIT |

| STEPS | Algorithm to delete an element from the circular queue |
|---|---|
| 1 | IF FRONT = -1<br>Write " UNDERFLOW "<br>Goto Step 4<br>[END of IF] |
| 2 | SET VAL = QUEUE[FRONT] |
| 3 | IF FRONT = REAR<br>SET FRONT = REAR = -1<br>ELSE<br>IF FRONT = MAX -1<br>SET FRONT = 0<br>ELSE<br>SET FRONT = FRONT + 1<br>[END of IF]<br>[END OF IF] |
| 4 | Exit |

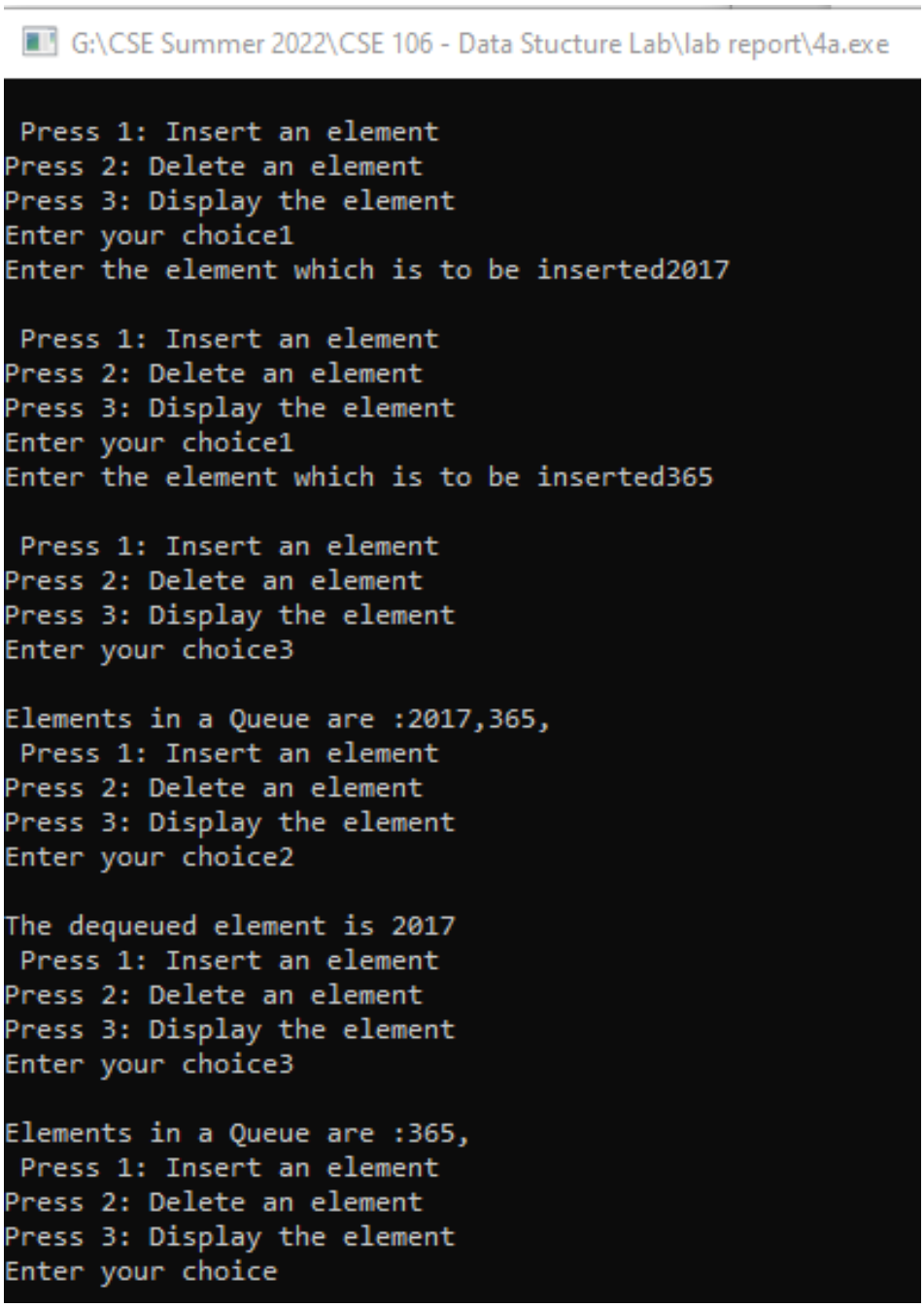
## 4. IMPLEMENTATION & TEST RESULT

### *Problem 1: circular queue*

```c
#include <stdio.h>
# define max 10
int queue[max];
int front=-1;
int rear=-1;
void enqueue(int element) {
    if(front==-1 && rear==-1) {
        front=0;
        rear=0;
        queue[rear]=element;
    } else if((rear+1)%max==front) {
        printf("Queue is overflow..");
    } else {
        rear=(rear+1)%max;
        queue[rear]=element;
    }
}
int dequeue() {
    if((front==-1) && (rear==-1)) {
        printf("\nQueue is underflow..");
    } else if(front==rear) {
        printf("\nThe dequeued element is %d", queue[front]);
        front=-1;
        rear=-1;
    } else {
        printf("\nThe dequeued element is %d", queue[front]);
        front=(front+1)%max;
    }
}
void display() {
    int i=front;
    if(front==-1 && rear==-1) {
        printf("\n Queue is empty..");
    } else {
        printf("\nElements in a Queue are :");
        while(i<=rear) {
            printf("%d,", queue[i]);
            i=(i+1)%max;
        }
    }
}
int main() {
    int choice=1,x;
    while(choice<4 && choice!=0) {
        printf("\n Press 1: Insert an element");
        printf("\nPress 2: Delete an element");
        printf("\nPress 3: Display the element");
        printf("\nEnter your choice");
        scanf("%d", &choice);
        switch(choice) {
            case 1:
                printf("Enter the element which is to be inserted");
                scanf("%d", &x);
                enqueue(x);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
        }
    }
    return 0;
}
```

## 4. IMPLEMENTATION & TEST RESULT

***Problem 1: circular queue***



## 1. TITLE OF THE LAB EXPERIMENT

Write a C program to for converting an infix expression to postfix expression using stack.

## 2. OBJECTIVES

Implement Stack.

## 3. PROCEDURE/ ANALYSIS / DESIGN

***Problem 1: Implement a program for converting an infix expression to postfix expression using stack.***

```c
#include<stdio.h>
#include<ctype.h>

char stack[100];
int top = -1;

void push(char x) {
    stack[++top] = x;
}

char pop() {
    if(top == -1)
        return -1;
    else
        return stack[top--];
}

int priority(char x) {
    if(x == '(')
        return 0;
    if(x == '+' || x == '-')
        return 1;
    if(x == '*' || x == '/')
        return 2;
    return 0;
}
```

```c

int main() {
    char exp[100];
    char *e, x;
    printf("Enter the expression : ");
    scanf("%s",exp);
    printf("\n");
    e = exp;

    while(*e != '\0') {
        if(isalnum(*e))
            printf("%c ",*e);
        else if(*e == '(')
            push(*e);
        else if(*e == ')') {
            while((x = pop()) != '(')
                printf("%c ", x);
        } else {
            while(priority(stack[top]) >= priority(*e))
                printf("%c ",pop());
            push(*e);
        }
        e++;
    }

    while(top != -1) {
        printf("%c ",pop());
    }
    return 0;
}

```

```
G:\CSE Summer 2022\CSE 106 - Data Stucture Lab\lab report\4b.exe
Enter the expression : a-b/c*d

a b c / d * -
--------------------------------
Process exited after 72.76 seconds with return value 0
Press any key to continue . . .
```

```
G:\CSE Summer 2022\CSE 106 - Data Stucture Lab\lab report\4b.exe
Enter the expression : (a+b)*c+(d-a)

a b + c * d a - +
--------------------------------
Process exited after 29.77 seconds with return value 0
Press any key to continue . . .
```

```
G:\CSE Summer 2022\CSE 106 - Data Stucture Lab\lab report\4b.exe
Enter the expression : ((4+8)(6-5))/((3-2)(2+2))

4 8 + 6 5 - 3 2 - 2 2 + /
--------------------------------
Process exited after 9.55 seconds with return value 0
Press any key to continue . . .
```

## 6. ANALYSIS AND DISCUSSION

1) This problem is solved by using c program. In this program we implement circular queue.

## 7. SUMMARY

1. We done this problem in c programming language.