**Assignment 6**

**Question 1**

Write a program implementing insert, delete and display operation of Circular Queue.

Ans:

#include<stdio.h>

# define MAX 5

int cqueue\_arr[MAX];

int front = -1;

int rear = -1;

void insert(int item)

{

if((front == 0 && rear == MAX-1) || (front == rear+1))

{

printf("Queue Overflow n");

return;

}

if(front == -1)

{

front = 0;

rear = 0;

}

else

{

if(rear == MAX-1)

rear = 0;

else

rear = rear+1;

}

cqueue\_arr[rear] = item ;

}

void deletion()

{

if(front == -1)

{

printf("Queue Underflown");

return ;

}

printf("Element deleted from queue is : %dn",cqueue\_arr[front]);

if(front == rear)

{

front = -1;

rear=-1;

}

else

{

if(front == MAX-1)

front = 0;

else

front = front+1;

}

}

void display()

{

int front\_pos = front,rear\_pos = rear;

if(front == -1)

{

printf("Queue is emptyn");

return;

}

printf("Queue elements :n");

if( front\_pos <= rear\_pos )

while(front\_pos <= rear\_pos)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

else

{

while(front\_pos <= MAX-1)

{

printf("%d ",cqueue\_arr[front\_pos])

front\_pos++;

}

front\_pos = 0;

while(front\_pos <= rear\_pos)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

}

printf("n");

}

int main()

{

int choice,item;

do

{

printf("1.Insertn");

printf("2.Deleten");

printf("3.Displayn");

printf("4.Quitn");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1 : printf("Input the element for insertion in queue : ");

scanf("%d", &item);

insert(item);

break;

case 2 : deletion();

break;

case 3: display();

break;

case 4: break;

default: printf("Wrong choicen");

}

}

while(choice!=4);

return 0;

}

**Question 2**

A Barua number is a number which consists of only zeroes and ones and has only one 1. Barua number will start with 1. Given numbers, find out the multiplication of the numbers. Note: The input may contain one decimal number and all other Barua numbers. (Assume that each number is very large and total number of values give is also very large)

Input 1: 100 10 12 1000

Output 1: 12000000

Input 2: 100 121 1000000000000000

Output 2: 12100000000000000000

Input 3: 10 100 1000

Output 3: 1000000

Ans:

* The only thing that we have to count the number of zeros in each of the numbers and simultaneously delete the zero in that number.
* So now we have numbers with no trailing zeros and a count variable with the number of zeros in every number.
* The counting is done by using divide(/) and modulus(%) with 10 and also using loops(mainly while loop is used).
* Finally multiple the numbers with no trailing zeros and using a loop print zeros(as such as the number in count variable) at the end.

while(n%10==0)

{

c++;

n=n/10;

}

p=1;

p=p\*n;

**Question 3**

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

#include <stdio.h>

#include <stdlib.h>

int data[100000],top,min=0;

void push()

{

   int item;

   scanf("%d",&item);

   top++;

   data[top]=item;

   if(min > data[top])

        min = data[top];

}

void pop()

{

    int i;

    if(min == data[top])

        min=0;

    top--;

    for(i= top;i>=0;i--)

        if(min > data[i])

            min = data[i];

}

int main()

{

    int t,n,choice;

    top = -1;

    scanf("%d",&t);

    while(t--)

    {

        scanf("%d",&choice);

        switch(choice)

        {

            case 1 : push();

                     break;

            case 2:  pop();

                     break;

            case 3: printf("%d\n",min);

                    break;

        }

    }

    return 0;

}