Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Data Structures Lab

Assignment No: 4

Name	: Gourav Balaji Suram
Roll No	: 39
PRN No	: 12220032

Problem Statement : Write a program to implement circular double ended queue where user can add and remove the element from both front and rear of the queue.

Program

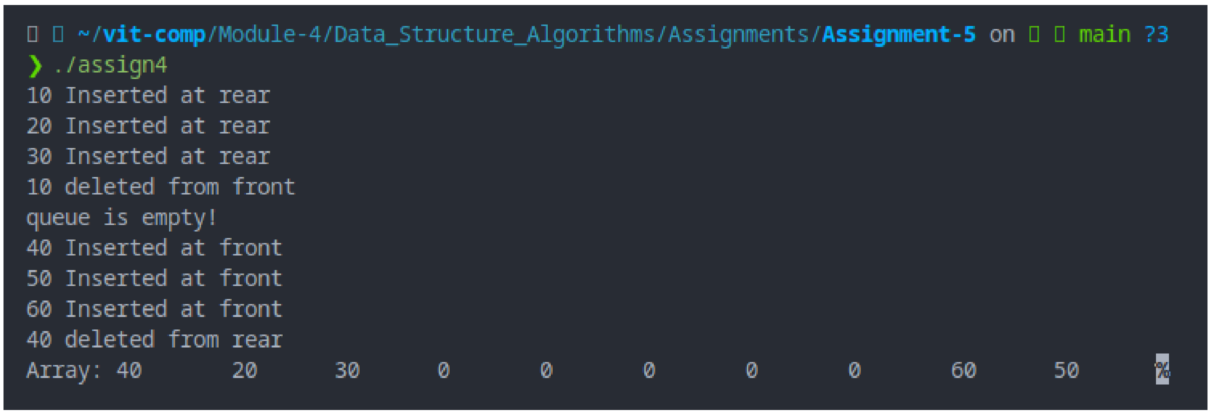
```
#include <stdio.h>

#define SIZE 10

int queue[SIZE];

int front = -1, rear = -1;

void insert_at_rare_end(int data){

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

printf("\nQueue is Full");

else {

if (front == -1) {

front = 0;

rear = 0;

        }

else if (rear == SIZE - 1){

rear = 0;

} else {

rear = rear + 1;

}

queue[rear] = data;

printf("%d Inserted at rear\n", data);

}

}

void insert_at_front_end(int data) {

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

printf("\nQueue is Full");
```

```c
else {
if (front == -1) {
front = 0;
rear = 0;
}
else if (front == 0) {
front = SIZE - 1;
}
else {
front = front - 1;
}
queue[front] = data;
printf("%d Inserted at front\n", data);
}
}
void delete_at_front(){
if (front == -1){
printf("queue is empty!\n");
} else{
printf("%d deleted from front\n", queue[front]);
if (front == rear){
front = rear = -1;
}
else if (front == SIZE - 1) {
front = 0;
} else {
front = front - 1;
```

```
        }
    }
}
        void delete_at_rear(){
        if (front == -1){
        printf("queue is empty!\n");
        }else{
        printf("%d deleted from rear\n", queue[rear]);
        if (front == rear){
        front = rear = -1;
        }
        else if (rear == 0){
        rear = SIZE - 1;
        }else{
        rear = rear - 1;
        }
    }
    }
        int main(){
        insert_at_rare_end(10);
        insert_at_rare_end(20);
        insert_at_rare_end(30);
        delete_at_front();
        delete_at_rear();
        insert_at_front_end(40);
        insert_at_front_end(50);
        insert_at_front_end(60);
```

```c
delete_at_rear();

printf("Array: ");

for (int i = 0; i < SIZE; i++){

printf("%d\t", queue[i]);

}

return 0;

}
```

OUTPUT



```
  ~/vit-comp/Module-4/Data_Structure_Algorithms/Assignments/Assignment-5 on   main ?3
❯ ./assign4
10 Inserted at rear
20 Inserted at rear
30 Inserted at rear
10 deleted from front
queue is empty!
40 Inserted at front
50 Inserted at front
60 Inserted at front
40 deleted from rear
Array: 40       20      30      0       0       0       0       0       60      50      %
```