Assignment - 3

Name :- G. Harshitha

Reg No :- 192324250
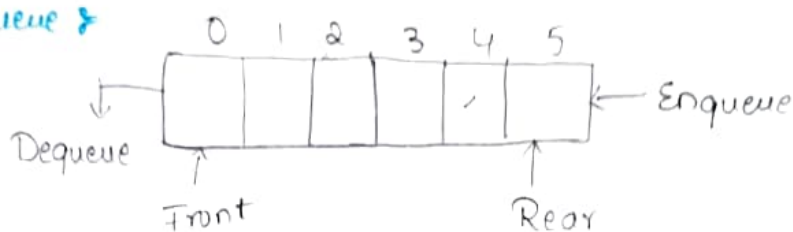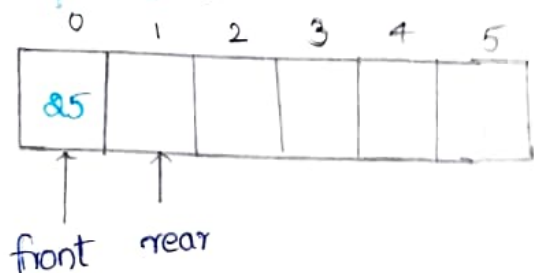
Department :- CSE(AI & DS)

Course Code :- CSA0389

Course Name :- Data Structure

Illustimate the queue operation using following function calls of size 5 enqueue(25), enqueue(37), enqueue(90) Dequeue( ), enqueue(15), enqueue(40), enqueue(12), dequeue(1). Dequeue( ).
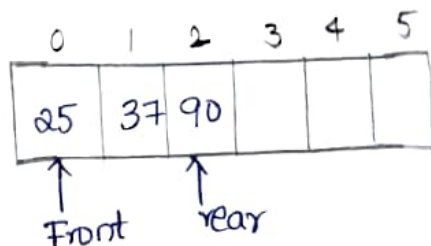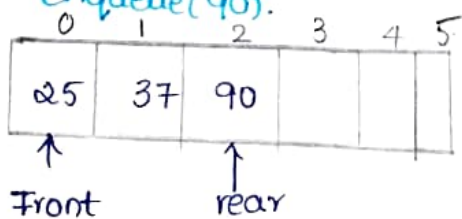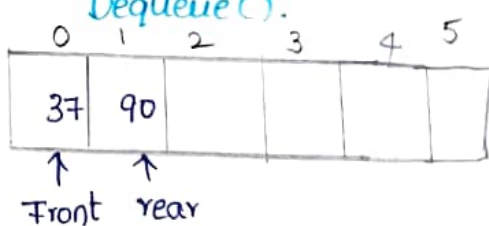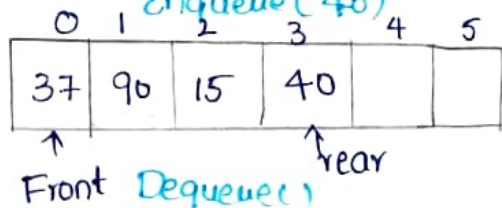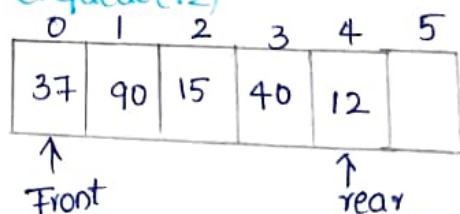
Queue :

Enqueue(25)

Enqueue(32)

Enqueue(90).

Dequeue( ).

Enqueue (40)

Front Dequeue( )

Enqueue(12)

queue (90)

Dequeue( ).

Dequeue(15)

```
 0   1   2   3   4   5
┌───┬───┬───┬───┬───┬───┐
│12 │   │   │   │   │   │
└───┴───┴───┴───┴───┴───┘
```
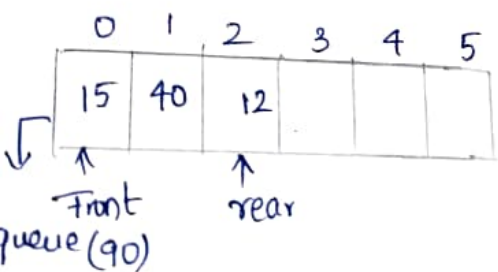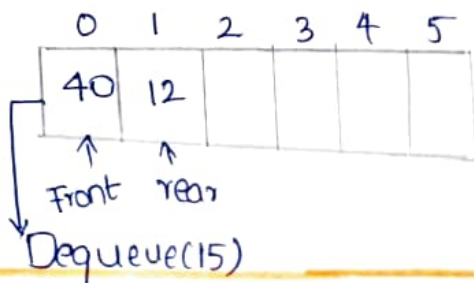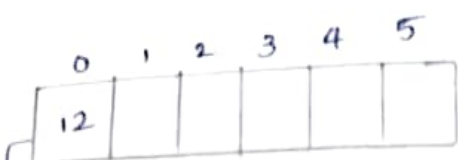
Dequeue(40)

rite a program to implement queue operation such

enqueue, display, dequeue.

```c
#include <stdio.h>
#include <stdlib.h>
# define < max[ ]

type def struct{
        int items[mAx];
        int front; rear; }
        queue
void initialize (queue *q){
        q -> front = -1;
        q -> rear = -1; }
void enqueue (queue *q, int value){
        if ( is full (q)){
        Printf("queue is full ! \n");
        return;
        }
    if (q -> front = -1){
        q -> front = 0
        q -> items[++q -> rear] = value;
        Printf("%d enqueued to queue \n", value);
        }
```

```c
void dequeue (queue *q){
    if (isempty(q)){
        Printf ("( queue is empty !\n");
        return;
    }
    Printf("%d dequed from queue \n", q->items[q->front +1]
}
void display( queue *q){
    if (is empty(q)){
        Printf("queue is empty !\n");
        return;
    }
    Printf("queue elements are : ");
    for(int i=q->front, i<=q->rear; i++){
        Printf ("%d; q->items[i]);
    }
    Printf ("\n");
}
int main(){
    Queue q;
    initialize (&q);
    Enqueue ()
    dequeue()
    display()
    return 0; }
```