

Assignment-3

Name ÷ CH. Naga Anjali

Regno ÷ 192372201

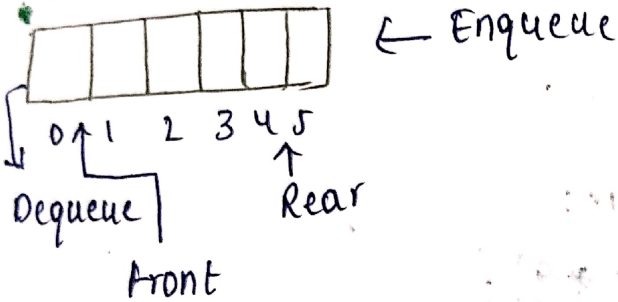
Dept ÷ CSE-AI

Course code ÷ CSA0389

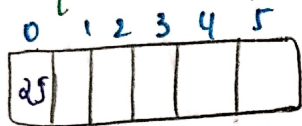
Course name ÷ Data Structures for stack
Overflow

Illustrate the Queue operation using following function.
 Calls of Size = 5, Enqueue(25), Dequeue(37), Enqueue(90),
 Dequeue(), Enqueue(15), Enqueue(40), Enqueue(12), Dequeue(),
 Dequeue(), Dequeue(), Dequeue()

Queue :-

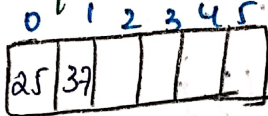


Enqueue(25) :-



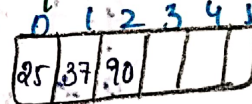
↑↑ Rear
Front

Enqueue(37) :-



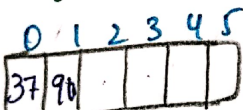
↑↑
Front Rear

Enqueue(90) :-



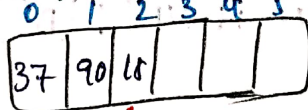
↑↑
Front Rear

Dequeue() :-



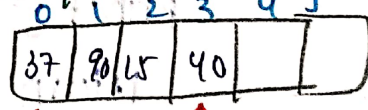
↑↑ Rear
Front

Enqueue(15) :-



↑↑
Front Rear

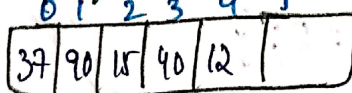
Enqueue(40) :-



↑↑
Front Rear

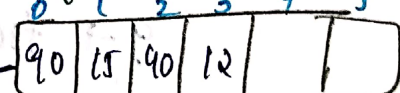
→ Dequeue(25)

Enqueue(12) :-



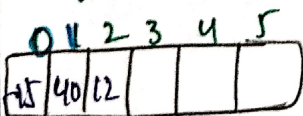
↑↑
Front Rear

Dequeue() :-



↑↑
Front Rear

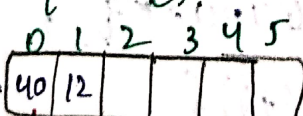
Dequeue() :-



↑↑
Front Rear

Dequeue(90)

Dequeue() :-



↑↑
Front Rear

Dequeue(15)

Dequeue(37)

Dequeue() :-



↑↑
Front Rear

Dequeue(40)

2) Write a C program to implement Queue operations such as Enqueue, Dequeue and Display.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5

typedef 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 ("val enqueued to queue \n", value)
}

void dequeue (Queue *q) {
    if (is empty (q)) {
        printf ("Queue is empty! \n");
        return ;
    }
}
```



```

    }
    printf (".d dequeued from Queue\n", q->item(q->front))
    }
    void display (Queue *q) {
        if (is empty (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;
    }

```