ASSIGNMENT - 3

NAME: Bhuvaneshwari. S

REG No: 192311375

COURSE CODE: CSA0389

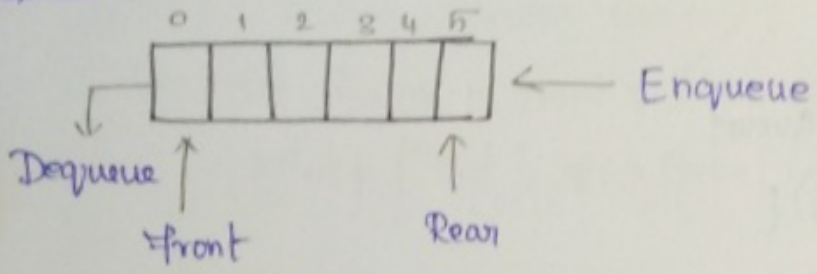
COURSE NAME: DATA STRUCTURE FOR STACK OVERFLOW

Faculty NAME: Dr. Ashok Kumar

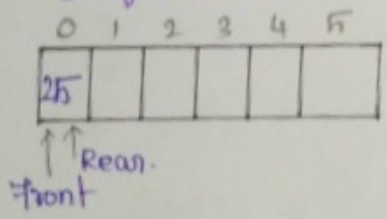
Submission Date: 21/08/24

Illustrate the queue operation using following function calls of size = 5. Enqueue (25), Enqueue (37), Enqueue (90), Dequeue(), Enqueue (15), Enqueue (40), Enqueue(12), Dequeue(), Dequeue(), Dequeue(), Dequeue().

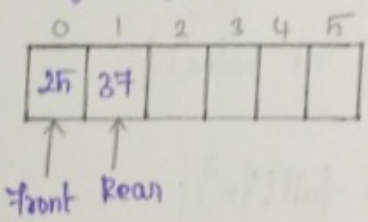
Queue:



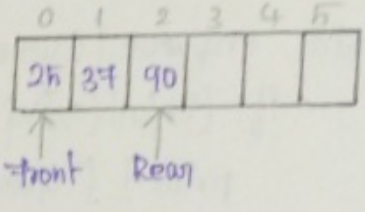
Enqueue (25):



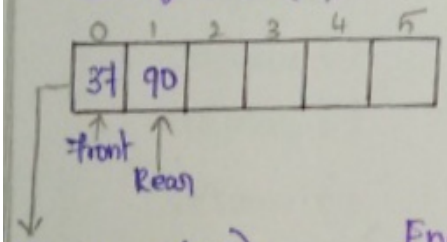
Enqueue (37):



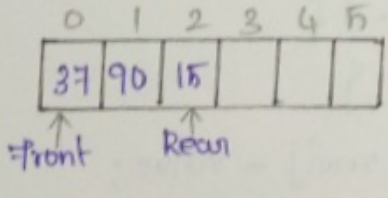
Enqueue (90):



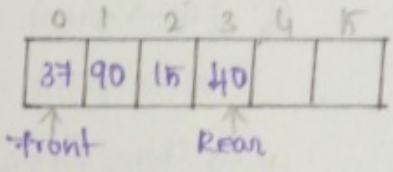
Dequeue ( ):



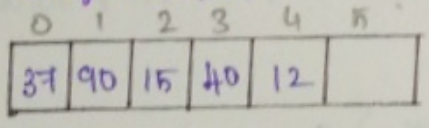
Enque(15):



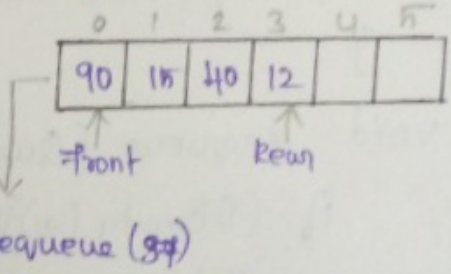
Enque(40):



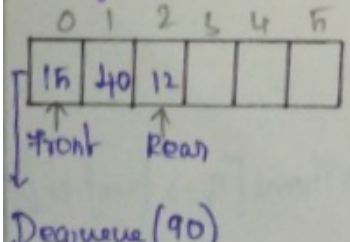
Enqueue (12):



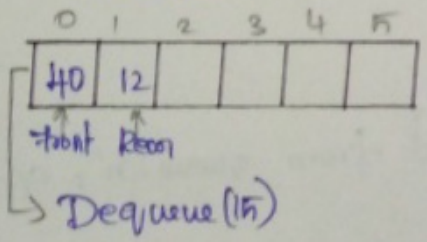
Dequeue ( ):



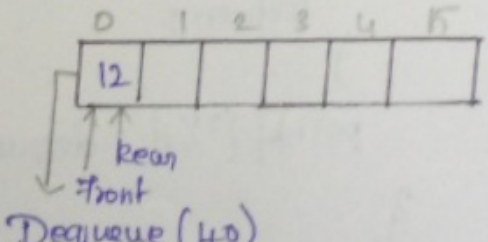
Dequeue ( )



Dequeue ( )



Dequeue ( ):



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 (isfull(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);
  }
void dequeue (Queue *q){
    if (Qisempty (q)){
      printf ("Queue is empty!\n");
      return;
    }
    printf ("%d dequeued from queue\n", q->items[q->front++]);
}
```

```
void display (Queue *q) {
    if (isEmpty (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]);
    }
    print f ("\n");
}
int main () {
    Queue q;
    initialize (&q);
    enqueue ()
    dequeue ()
    display ()
    return 0;
}
```