Assignment - 3

Name : R. Nikhila

Register Number : 192372186

course Name : DataStructure

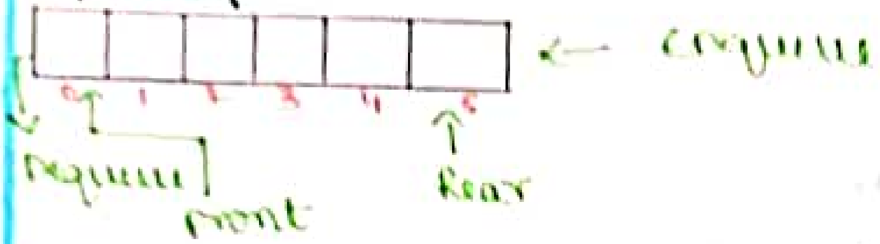
course code : CSA0389

Submission date : 21-08-2024

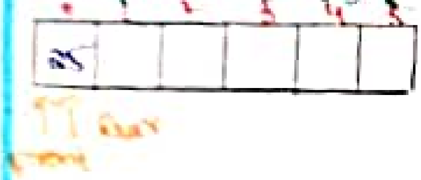
Department : CSE- AI

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()

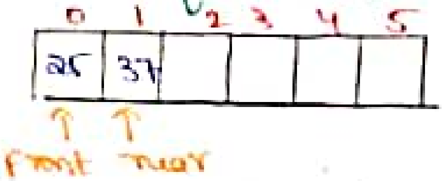
Queue:



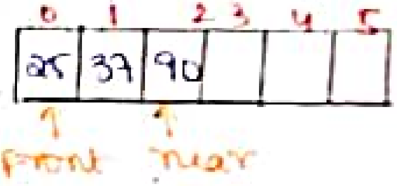
Enqueue(25):



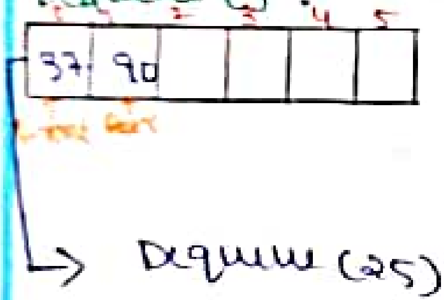
Enqueue(37):



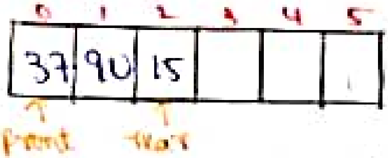
Enqueue(90):



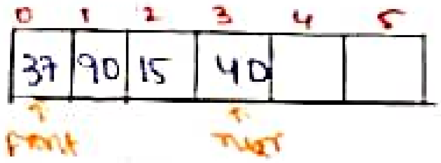
Dequeue():



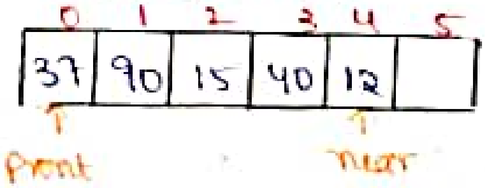
Enqueue(15):



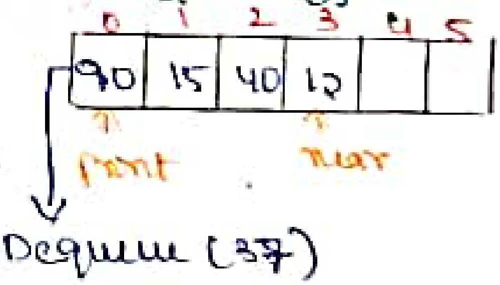
Enqueue(40):



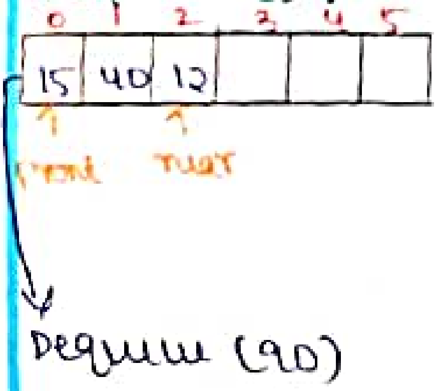
Enqueue(12):



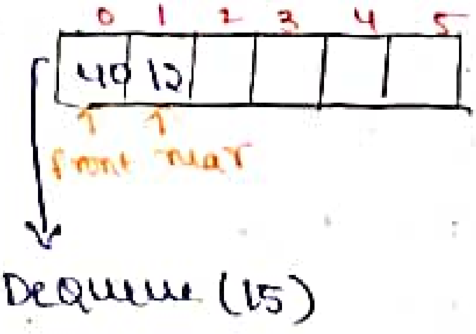
Dequeue():



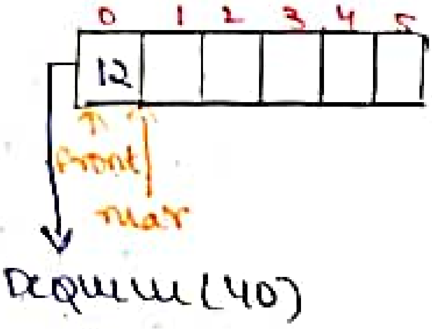
Dequeue():



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 enqued to queue \n", value); }
void dequeue (queue *q) {
     if (isempty (q)) {
   printf (" queue is empty !\n");
    return;
  }
printf(" %d dequeued from queue \n", q-> items[q->front]);
  }

 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;
    }
```