CSA-0389

Datastructure

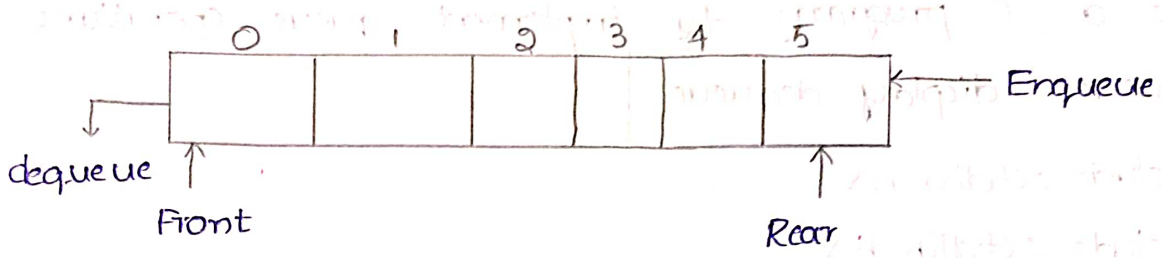
for stackoverflow

Assignment-3

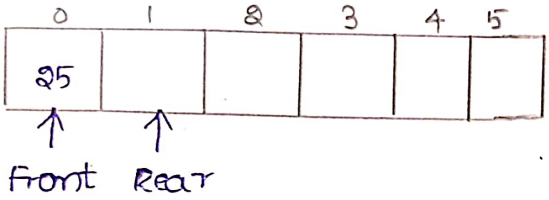
B.MALLIKA
192324234
AI&DS

Illustrate the queue operation using following function calls of size = 5 enqueue (25), enqueue (37), enqueue (90), Dequeue(), enqueue (15), enque (40), enqueue (12), deque(1), dequeue()

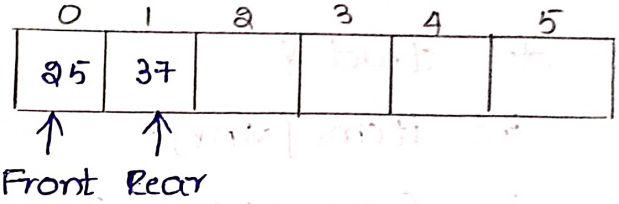
Queue



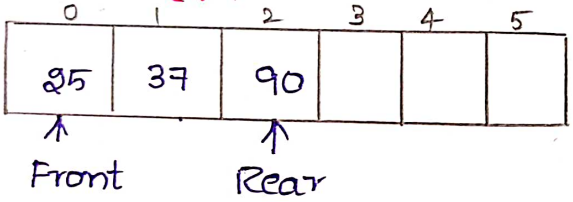
Enqueue (25)



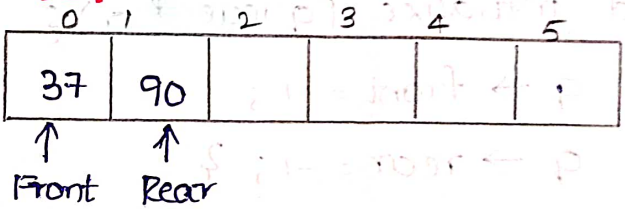
Enqueue (37)



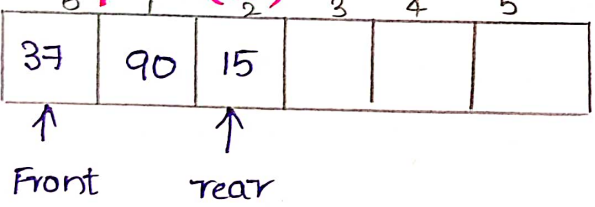
Enqueue (90)



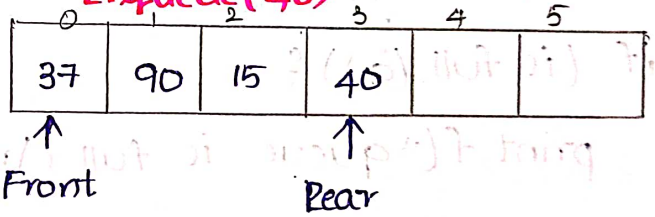
Dequeue ()



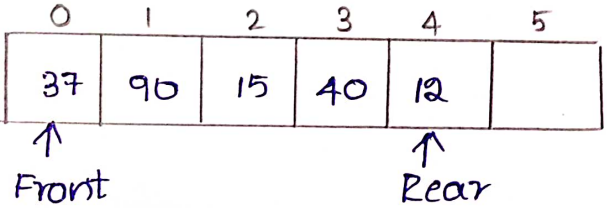
Enqueue (15)



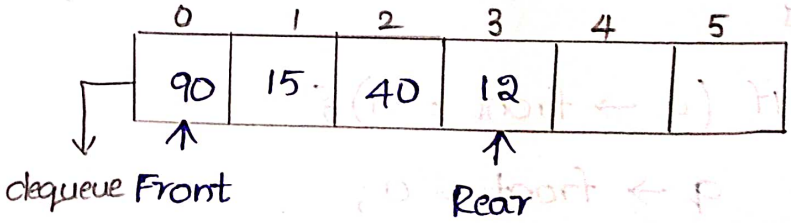
Enqueue (40)



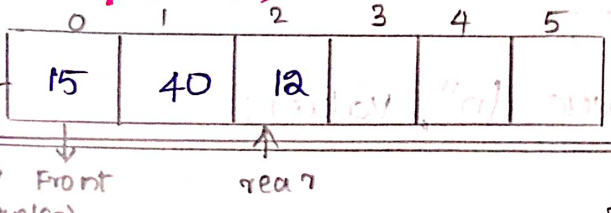
Enqueue (12)



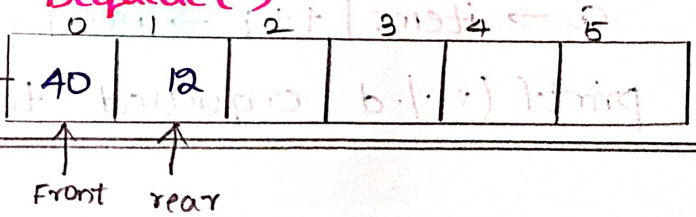
Dequeue ()

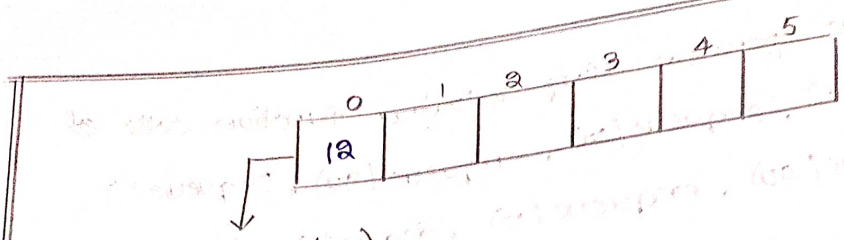


Dequeue ()



Dequeue ()





dequeue (40);

2. Write a C program to implement queue operations such as enqueue, display dequeue.

```
#include <stdio.h>
#include <stdlib.h>
#define <max 5>
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)) {
     print f("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 (isempty (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]);
}
  printf ("\n");
}
  int main() {
      Queue q;
      Initialize (&q);
      Enqu...
```

```
    display ( )
    return 0;
}
```