

Ex. No. **Developing and executing 'C' program to implement Queue operation - Enqueue**

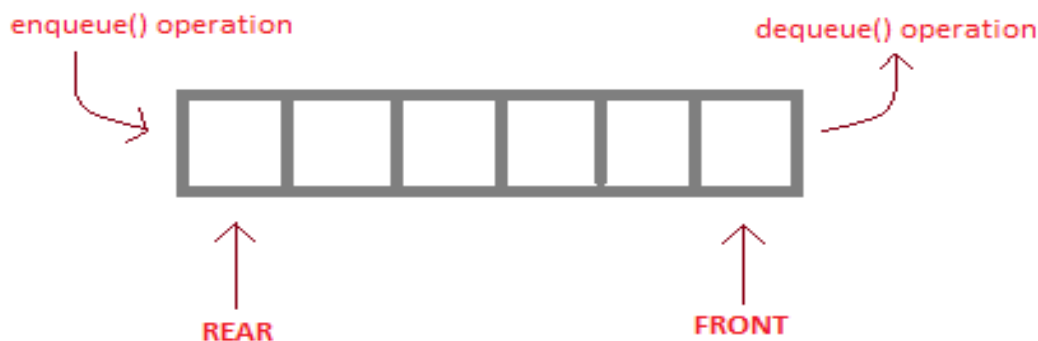
Date :

QUEUE

- ☐ Queues are linear data structures in which elements can be inserted in one end and deleted from the other end
- ☐ This is referred to as a **first-in-first-out (FIFO) system**
- ☐ The element which is inserted first will be the first to be removed. A new element is inserted as the last element of the queue.

Operations Performed On Queue

- ☐ FRONT - The end at which the elements are deleted
- ☐ REAR - The end at which the elements are inserted
- ☐ ENQUEUE - Insertion of elements into a queue
- ☐ DEQUEUE - Deletion of elements from a queue



enqueue() is the operation for adding an element into Queue.

dequeue() is the operation for removing an element from Queue .

QUEUE DATA STRUCTURE

Exercise : Write a 'C' program to implement Queue operation

```
/* To implement Queue operation */
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 5
int queue[MAX],front=0,rear=0;
void insert( );
void delete( );
void display( );
void main( )
```

```

{
    int ch;
    clrscr( );
    while(1)
    {
        printf("\n-----QUEUE OPERATIONS          \n");
        printf("\n 1.Insert \n 2.Delete \n 3.Display \n 4.Exit\n");
        printf("\n\n Enter your choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1: insert();
                    break;
            case 2: delete();
                    break;
            case 3: display();
                    break;
            case 4: exit(0);
            default: printf("Invalid option\n");
        }/*switch*/
    }/*while*/
}/*main*/

```

/* Function for Insertion operation */

```

void insert( )
{
    int x;
    if(rear==MAX)
    {
        printf("QUEUE IS FULL\n");
    }
}

```

```

else
{
    printf("The Enter queue element is : ");
    scanf("%d",&x);
    queue[rear]=x;
    rear++;
}
}/*insert function */

```

/* Function for Deletion operation */

```

void delete( )
{
    int i;
    if(front==rear)
    {
        printf("QUEUE IS EMPTY\n");
    }
    else
    {
        printf("The deleted element is : %d",queue[front]);
        for(i=0;i<rear;i++)
        {
            queue[i] = queue[i+1];
        }
        rear--;
    }/*if */
}/*Delete function*/

```

/* Function for Display operation */

```

void display( )
{
    int i;
    if(front==rear)
    {
        printf("QUEUE IS EMPTY");
    }
    else
    {
        printf("queue elements are : \t");
        for(i=front;i<rear;i++)
        {
            printf("%d\n",queue[i]);
        }
    }/*if*/
} /*Display function */

```

