



# Data Structures : Queue

B.Sc. 3<sup>rd</sup> Semester, Paper C5

Paulami Basu Ray

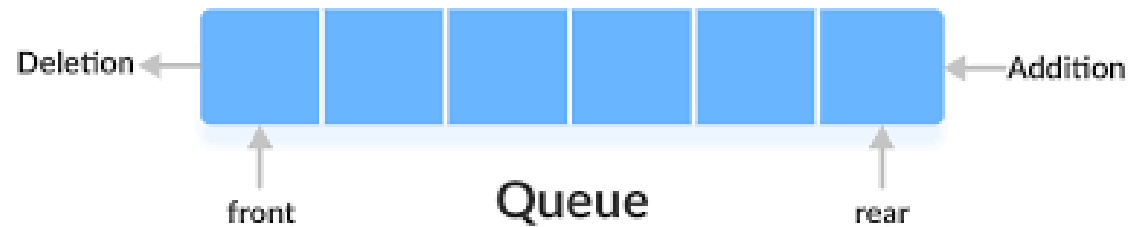
Assistant Professor

Department of Computer Science & Applications

Prabhat Kumar College, Contai

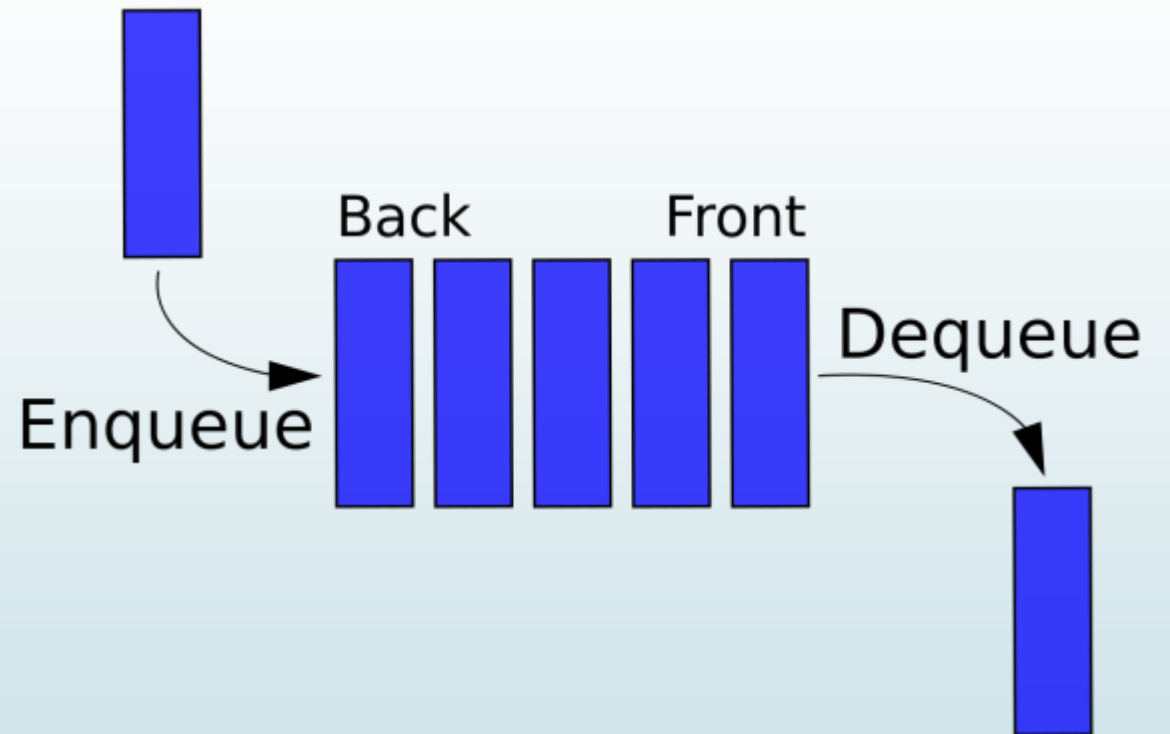
# Queue: Definition

- ▶ A Queue is a linear structure which follows a particular order in which the operations are performed.
- ▶ The order is First In First Out (FIFO).
- ▶ A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first.
- ▶ The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

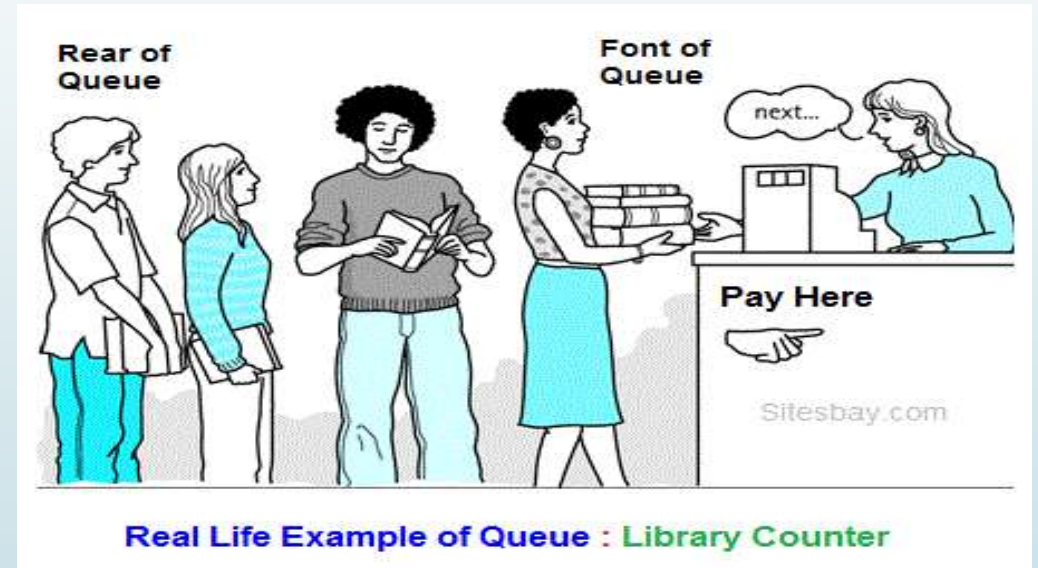


# Queue Operations

- ▶ Enqueue
- ▶ Dequeue



# Real Life Examples



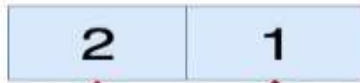
Empty



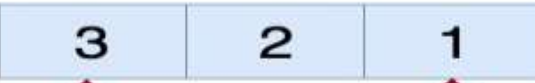
Enqueue 1



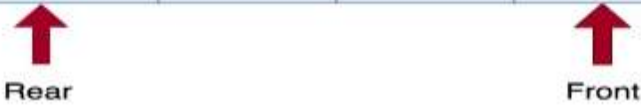
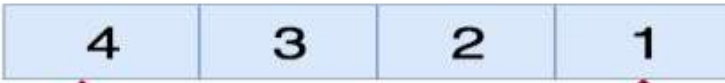
Enqueue 2

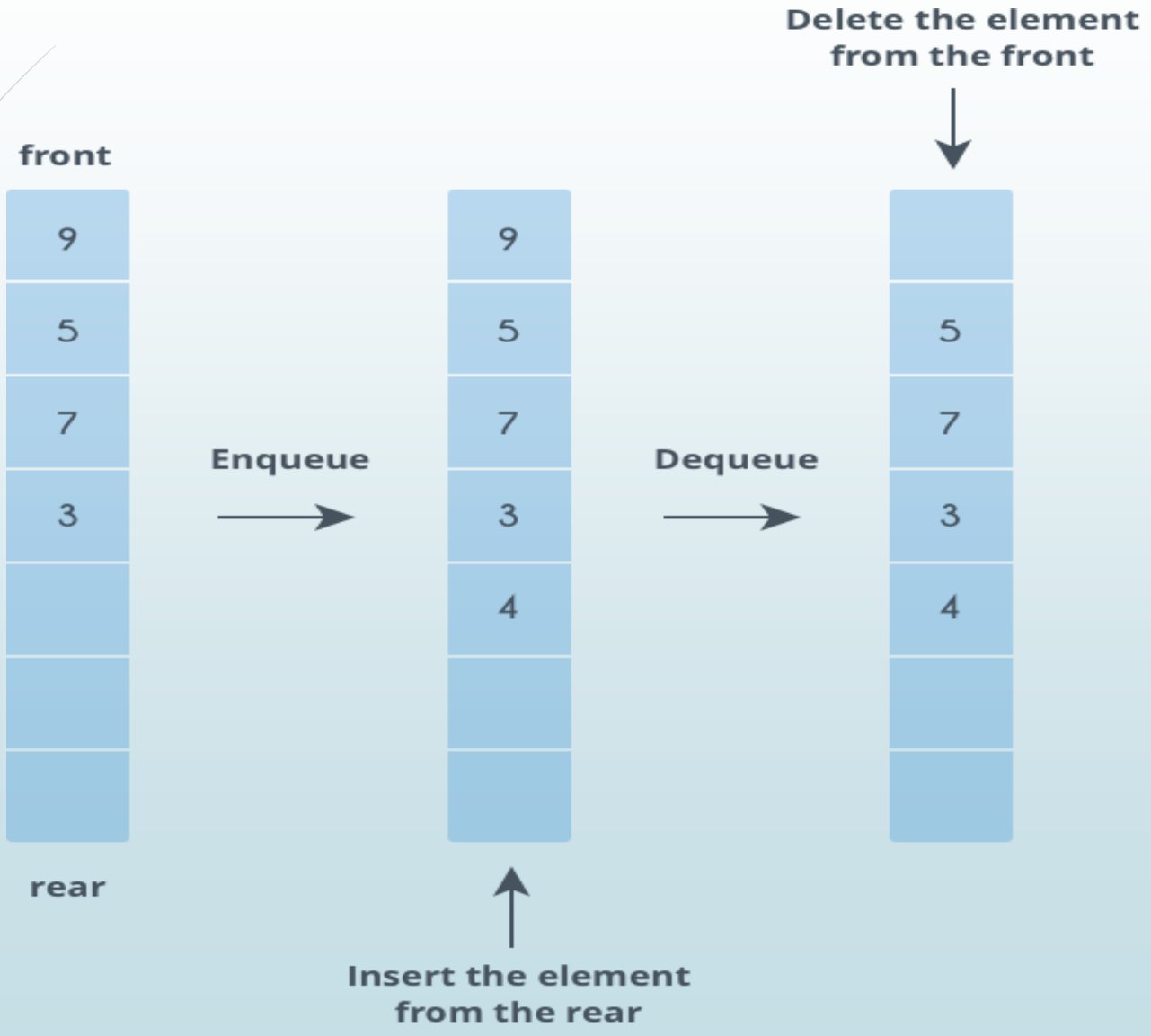


Enqueue 3



Enqueue 4







## C Program for Queue implementation

```
/*  
    C Program to Implement a Queue using an Array  
*/  
#include <stdio.h>  
  
#define MAX 50  
  
void enqueue();  
void dequeue();  
void display();  
int queue_array[MAX];  
int rear = - 1;  
int front = - 1;  
void main()  
{  
    int choice;  
    while (1)  
    {  
        printf("1.Insert element to queue \n");  
        printf("2.Delete element from queue \n");  
        printf("3.Display all elements of queue \n");  
        printf("4.Quit \n");  
        printf("Enter your choice:");  
        scanf("%d",&choice);
```

```
switch (choice)
{
    case 1:
        enqueue();
        break;
    case 2:
        dequeue();
        break;
    case 3:
        display();
        break;
    case 4:
        return;
    default:
        printf("Wrong choice \n");
} /* End of switch */
} /* End of while */
} /* End of main() */
```



```
void enqueue()
{
    int add_item;
    if (rear == MAX - 1)
        printf("Queue Overflow \n");
    else
    {
        if (front == - 1)
            /*If queue is initially empty */
            front = 0;
        printf("Inset the element in queue : ");
        scanf("%d", &add_item);
        rear = rear + 1;
        queue_array[rear] = add_item;
    }
} /* End of insert() */
```

```
void dequeue()
{
    if (front == - 1 || front > rear)
    {
        printf("Queue Underflow \n");
        return ;
    }
    else
    {
        printf("Element deleted from queue is : %d\n",
            queue_array[front]);
        front = front + 1;
    }
} /* End of dequeue() */
```

```
void display()
{
    int i;
    if (front == - 1)
        printf("Queue is empty \n");
    else
    {
        printf("Queue is : \n");
        for (i = front; i <= rear; i++)
            printf("%d ", queue_array[i]);
        printf("\n");
    }
} /* End of display() */
```