

## DATA STRUCTURE

### Ishwar Prakash (MCA)

**QUEUE:** 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 stack 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 is an abstract data structure, somewhat similar to Stacks. Unlike stacks, a queue is open at both its ends. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue). Queue follows First-In-First-Out methodology, i.e., the data item stored first will be accessed first.



A real-world example of queue can be a single-lane one-way road, where the vehicle enters first, exits first. More real-world examples can be seen as queues at the ticket windows and bus-stops.

### Program

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<process.h>

#define SIZE 5

int menu()
{
    int choice;

    clrscr();

    printf("===MENU===");

    printf("\n 1.INSERT");

    printf("\n 2.DELETE");

    printf("\n 3.DISPLAY");

    printf("\n 4.EXIT");

    printf("\n Input your choice");

    scanf("%d",&choice);

    return choice;
}

////////////////////////////////////

void main()
{
    int arr[SIZE],index,front=-1,rear=-1;
```

```
while(1)
{
    switch(menu())
    {
        case 1:
            if(rear==SIZE-1)
            {
                printf("Queue is Full");
            }
            else
            {
                rear=rear+1;
                printf("Input the Data");
                scanf("%d",&arr[rear]);
            }
            getch();
            break;
        case 2:
            if(front==rear||front==SIZE-1)
```

```
    {  
        printf("Queue is Empty");  
    }  
else  
    {  
        front=front+1;  
        printf("Deleted Item=%d",arr[front]);  
    }  
    getch();  
    break;  
case 3:  
    for(index=front+1;index<=rear;index++)  
    {  
        printf("\n %d",arr[index]);  
    }  
    getch();  
    break;  
case 4:  
    exit(1);
```

default:

```
printf("\n Invalid Choice");
```

```
getch();
```

```
}
```

```
}
```

```
}
```