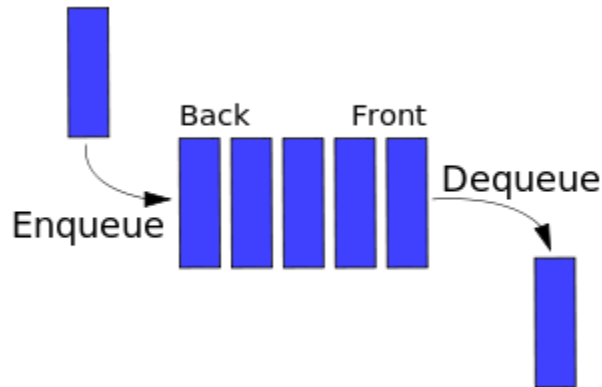


Fifo Queue

कंप्यूटर विज्ञान में, एक कतार संग्रह में संस्थाओं के आदेश और संग्रह पर कार्रवाई में जाना पीछे टर्मिनल स्थिति, के लिए संस्थाओं के अलावा हैं प्रमुख (या केवल) में रखा जाता है, जिसमें सार डेटा प्रकार या संग्रह की एक विशेष प्रकार है enqueue के रूप में, और विपंक्ति के रूप में जाना सामने टर्मिनल स्थिति, से संस्थाओं को हटाने की. इस पंक्ति में एक प्रथम में सबसे पहले आउट (एफआईएफओ) डेटा संरचना बनाता है. एक फीफो डेटा संरचना में पहला तत्व को हटाया जाना पहले से एक होगा कतार में जोड़ा. इस बार एक नया तत्व जुड़ जाता है कि आवश्यकता, नए तत्व हटाया जा सकता से पहले हटा दिया जाना है पहले जोड़ा गया था कि सभी तत्वों के बराबर हैं. अक्सर एक झांकना या सामने आपरेशन भी यह dequeuing बिना सामने तत्व का मूल्य लौटने कार्यान्वित किया जाता है. एक कतार एक रेखीय डेटा संरचना, या अधिक सूक्ष्म रूप में एक अनुक्रमिक संग्रह का एक उदाहरण है.



CODES [C]

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
typedef struct queue
{
    int *a ;
    int ms,r,f;
}Q;
void init(Q *q,int x)
{
    q->ms=x;
    q->a=(int *)malloc(q->ms*sizeof(int));
    q->f=0;
    q->r=-1;
}
int isfull(Q *q)
{
    if(q->r==q->ms-1&&q->f==0)
        return 1;
    else
        return 0;
}
int isempty(Q *q)
{
    if(q->r==-1&&q->f==0)
        return 1;
    else
        return 0;
}
void insert(Q *q,int z)
{
    if(isfull(q))
        printf("queue full ");
    else
        q->a[++q->r]=z;
}
int del(Q *q)
{
    int j=0;
    if(isempty(q)==1)
    {
        return 1 ;
    }
    else
    {
        j=q->a[q->f++];
        return j;
    }
}
```

CODES (JAVA)

```
import java.io.*;
class queue
{
    int q[];
    int front;
    int rear;
    int temp;
    int max;
    queue(int n)
    {
        max=n;
        q=new int[max];
        front=0;
        rear=-1;
    }
    void push(int a)
    {
        int b;
        b=a;
        if(rear>=max-1)
        {
            System.out.println("\t\t*****queue is overflow!!!!*****\n");
        }
        else
        {
            rear=rear+1;
            q[rear]=b;
        }
    }
}
```

```
void pop()
{
    if(rear<front)
    {
        System.out.println("\t\tstack is underflow!!!!\n");
    }
    else
    {
        System.out.println(""+q[front]+"is deleted\n");
        front=front+1;
    }
}
void display()
{
    System.out.println("\n\t\tElements are:");
    for(int i=front;i<=rear;i++)
    {
        System.out.println(""+ "\t\t"+q[i]);
    }
    System.out.println("\n");
}
}
```