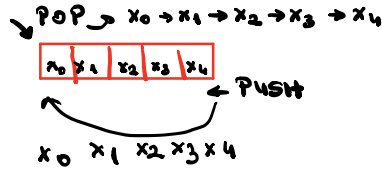


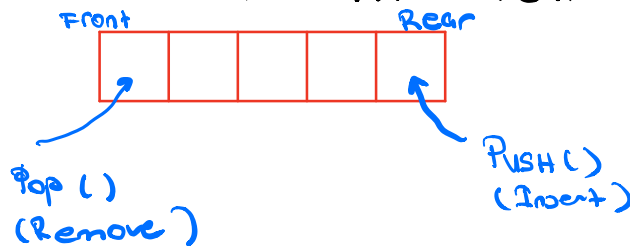
Queue: İki yönden harekete izin veren dizi yapısı:

First In First Out : FIFO

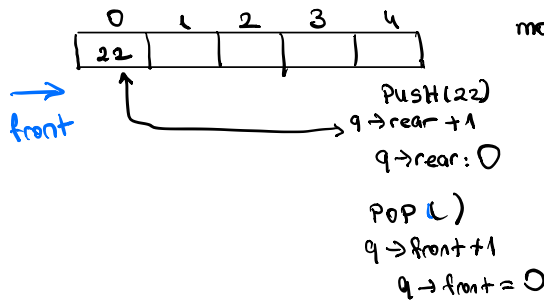


Silme işlemi ön taraftan : front

Ekleme işlemi arka taraftan : rear



Kodlama Tekniği: (Evrimsel kuyruk tekniği)



$mod\ S = mod\ SIZE$

$q \rightarrow front = SIZE - 1$

$q \rightarrow rear = SIZE - 1$

Operasyonlar:

① Is Empty / IsFull

② POP - (Remove)

③ PUSH (Insert)

Veri Yapısı:

```
struct queue
{
    double items [SIZE];
    int front;
    int rear;
}
```

Test : Kuyruk Boş mu ?

```
bool IsEmpty (struct *q)
{
    if ( q → front == q → rear )
```

```

    return true;
else
    return false;
}

```

Queue Ekleme: PUSH()

```

void insert ( struct queue *q , double x)
{ if ( (q->rear)+1) % SIZE == q->front ) → qift yonlu pointer kontrolu.
    { printf ("Kuyruk dolu %.1f elemanı eklenmedi! \n", x);
    }
else
    { q->rear = (q->rear + 1) % SIZE; → modular teknik
      q->items[q->rear] = x;
    }
}
}

```

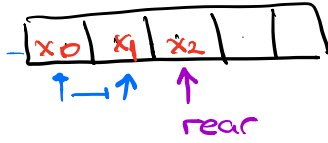
Queue Çıkarma : POP()

```

double remove ( struct queue *q)
{
    if ( ! isEmpty(q) )
    {
        q->front = (q->front + 1) % SIZE;
        double d = q->items[q->front];
        return d;
    }
    else
    { printf ("Kuyruk Boş , çıkarma yapılamaz \n" );
      return 0;
    }
}
}

```

$q \rightarrow \text{front} = \text{SIZE} - 1 = 4$ Arr



$q \rightarrow \text{rear} = \text{SIZE} - 1 = 4$

PUSH(x_0) \rightarrow $q \rightarrow \text{rear}++ \rightarrow 0$ $5 = 0 \bmod(5)$ Arr[0] = x_0

PUSH(x_1) \rightarrow $q \rightarrow \text{rear}++ \rightarrow 1$ $6 = 1 \bmod(5)$ Arr[1] = x_1

PUSH(x_2) \rightarrow $q \rightarrow \text{rear}++ \rightarrow 2$ $7 = 2 \bmod(5)$ Arr[2] = x_2

POP() \rightarrow $q \rightarrow \text{front}++ \rightarrow 0$ $5 = 0 \bmod(5)$ Arr[0] = Null

POP() \rightarrow $q \rightarrow \text{front}++ \rightarrow 1$ $6 = 1 \bmod(5)$ Arr[1] = Null

POP() \rightarrow - - - - -

Yazdırma

Version 1:

(struct queue q)

```
void print Queue (struct queue *q)
{ while (is Empty(q))  $\rightarrow$  sq
  { printf ("%f", remove(q));
  }
}
```

int main()

{ struct queue q; -

q.front = SIZE - 1;

q.rear = SIZE - 1;

double d = remove(sq);

Insert(sq, 0);

Insert(sq, 1);

d = remove (sq);

printf (" Çıktı Elemanı = %.1f \n", d);

printf (" Kuyruğun son durumu ! \n");

print Queue (q);

return 0;

}

Uygulama Dersi:

SORU1: Kuyruğun en büyük elemanını ve bu elemanın sırasını döndüren fonksiyonu yazınız.

SORU2: Kuyrukta ki elemanların ortalamasını alan fonksiyonu yazınız

SORU3: Kuyrukta ki en büyük elemanı en öne getiren ve sonucu döndüren fonksiyonu yazınız.

SORU4: İlk ve sonun yığıt (stack) iken de çözünüz!