به نام خدا

ساختمان داده ها و الگوریتم ها

محمدمهدی گیلانیان صادقی

دانشگاه آزاد اسلامی واحد قزوین

نیمسال دوم ۱۴۰۱-۱۴۰۲

- **صف (queue):**

مجموعه ای از عناصر مرتب است که هر عنصر از یک طرف به نام **ابتدای صف** از آن حذف و از طرف دیگر به نام **انتهای صف** به آن اضافه می شود.

صف را ساختمان داده FIFO (First In First Out) می نامند.

عملیات اصلی:

- ایجاد صف خالی
- بررسی خالی بودن صف
- *بررسی پر بودن صف*
- اضافه کردن عنصر به انتهای صف
- حذف کردن عنصر از ابتدای صف
- بازیابی عنصر از ابتدای صف

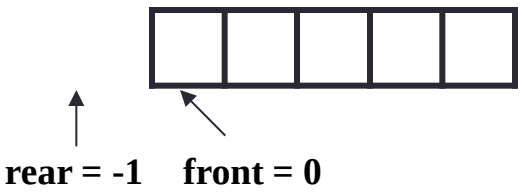
پیاده سازی صف:

- با استفاده از آرایه ها
- با استفاده از لیست های پیوندی

پیاده سازی صف با استفاده از آرایه ها:

مثال: از دو متغیر front و rear به ترتیب برای نگهداری ابتدا و انتهای صف استفاده می کنیم.

rear به عنصر انتهای صف اشاره می کند و در ابتدا rear= -1 است.
front به عنصر ابتدای صف اشاره می کند و در ابتدا front= 0 است.



پیاده سازی صف با استفاده از آرایه ها:

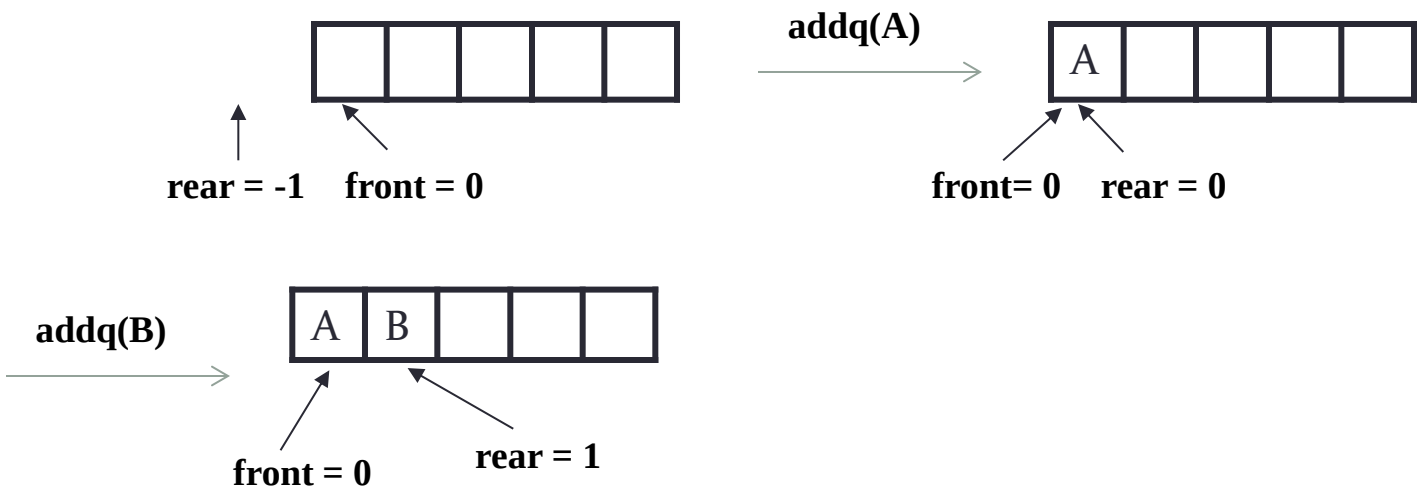
مثال: از دو متغیر front و rear به ترتیب برای نگهداری ابتدا و انتهای صف استفاده می کنیم.

rear به عنصر انتهای صف اشاره می کند و در ابتدا rear= -1 است.
front به عنصر ابتدای صف اشاره می کند و در ابتدا front= 0 است.

پیاده سازی صف با استفاده از آرایه ها:

مثال: از دو متغیر front و rear به ترتیب برای نگهداری ابتدا و انتهای صف استفاده می کنیم.

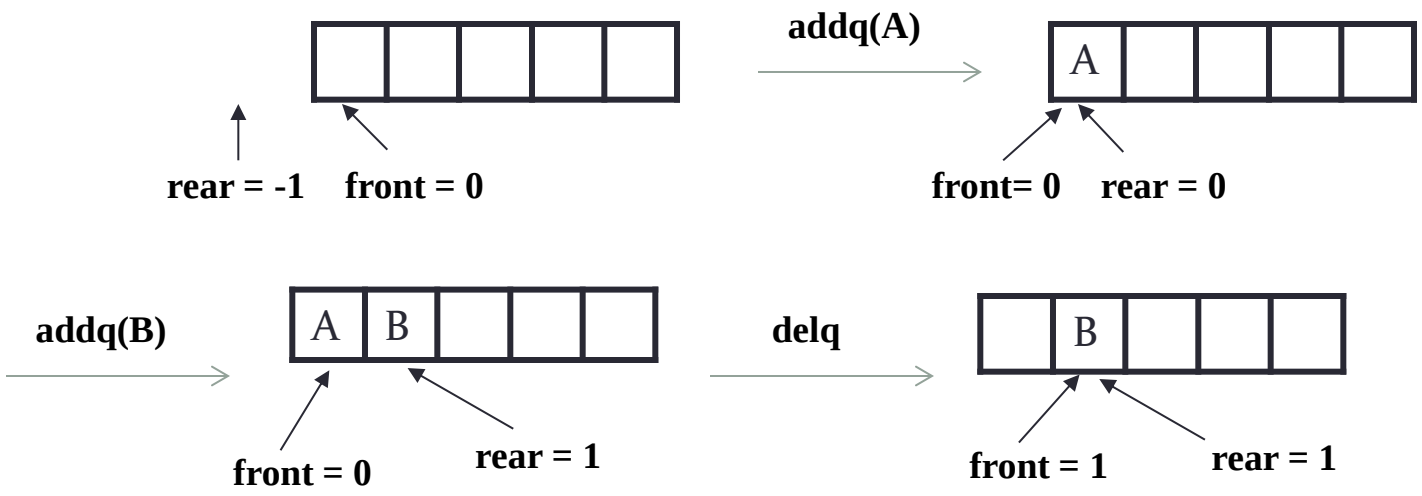
rear به عنصر انتهای صف اشاره می کند و در ابتدا rear= -1 است.
front به عنصر ابتدای صف اشاره می کند و در ابتدا front= 0 است.



پیاده سازی صف با استفاده از آرایه ها:

مثال: از دو متغیر front و rear به ترتیب برای نگهداری ابتدا و انتهای صف استفاده می کنیم.

rear به عنصر انتهای صف اشاره می کند و در ابتدا rear= -1 است.
front به عنصر ابتدای صف اشاره می کند و در ابتدا front= 0 است.



پیاده سازی کلاس صف:

```
#define SIZE 5
class queue{
 public:
    queue();
    int empty();
//  int full();
    void addq(int, int &);
    void delq(int &, int &);
    void retrieveq(int &, int &);
   // other member functions
 private:
    int front;
    int rear;
    int items[SIZE];
};
```

پیاده سازی عمل ایجاد صف:

```
queue::queue()
{
  front = 0;
  rear = -1;
}
```

پیاده سازی عمل ایجاد صف:

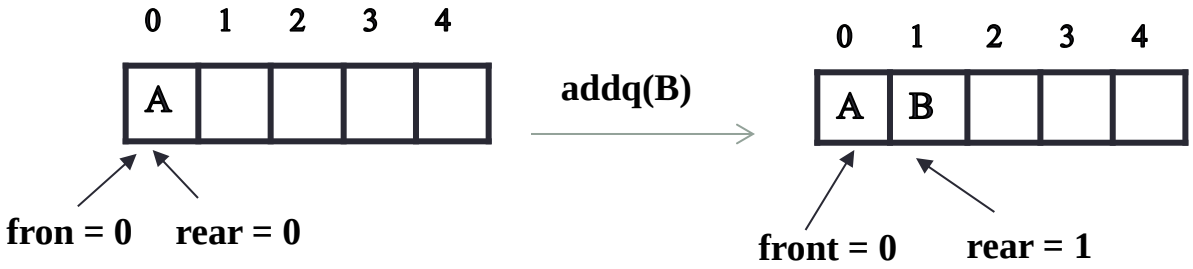
```
queue::queue()
{
  front = 0;
  rear = -1;
}
```

پیاده سازی عمل بررسی خالی بودن صف:

```
int queue::empty()
{
 if (rear < front)
   return 1;
 return 0;
}
```

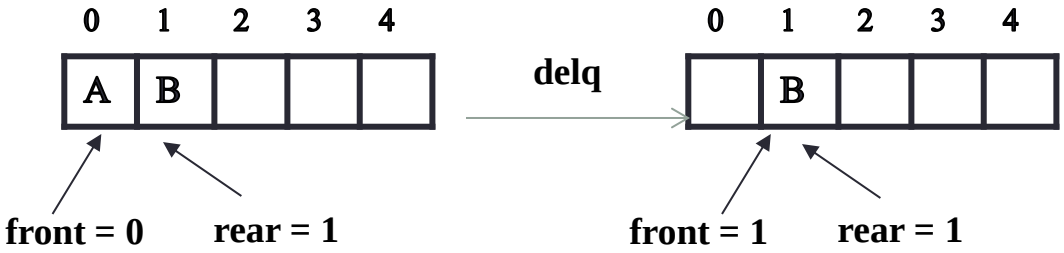
پیاده سازی عمل افزودن عنصر به انتهای صف:

```
void queue::addq(int x, int &overflow)
{
  if (rear==SIZE-1)
       overflow=1;
  else
    {
       overflow=0;
       items[++rear]=x;
    }
}
```



پیاده سازی عمل حذف عنصر از جلوی صف:

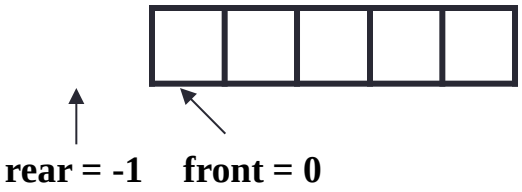
```
void queue::delq(int &x, int &underflow)
{
  if (empty ())
      underflow=1;
  else
    {
      underflow=0;
      x=items[front++];
    }
}
```



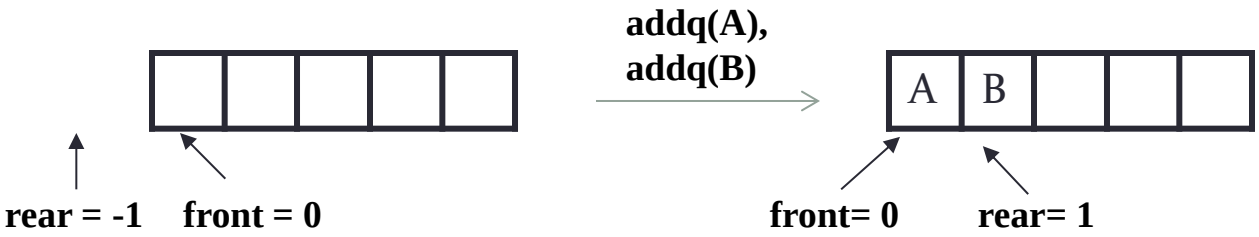
پیاده سازی عمل بازیابی عنصر از جلوی صف:

```
void queue::retrieveq(int &x, int &underflow)
{
  if (empty ())
      underflow=1;
  else
    {
       underflow=0;
       x=items[front];
    }
}
```

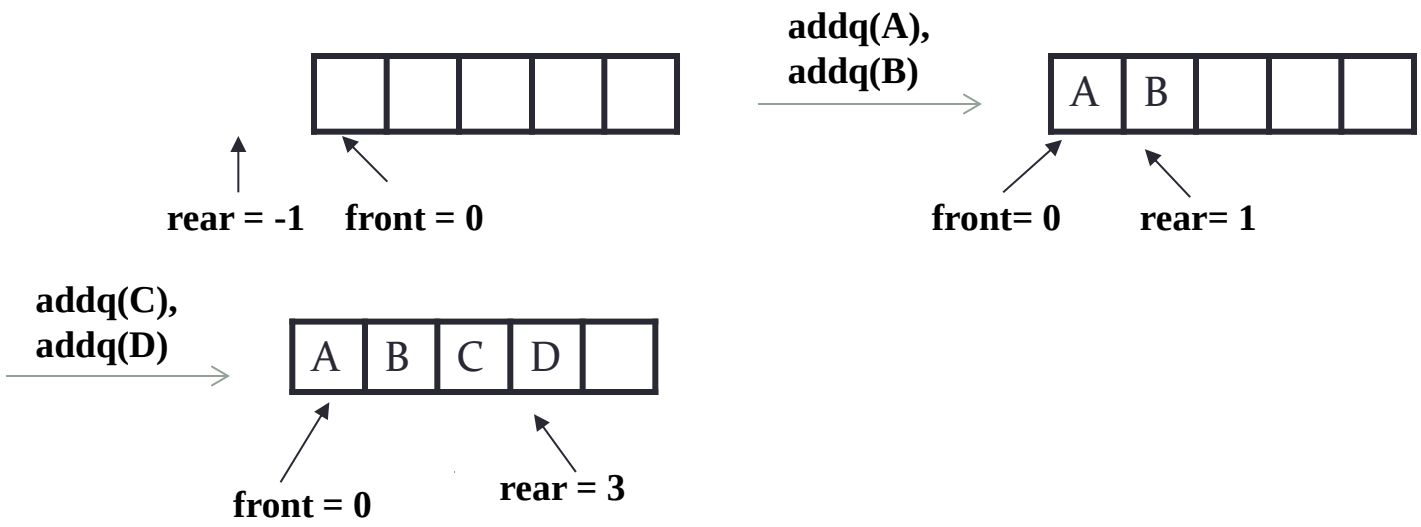
مشکلات پیاده سازی صف با آرایه:



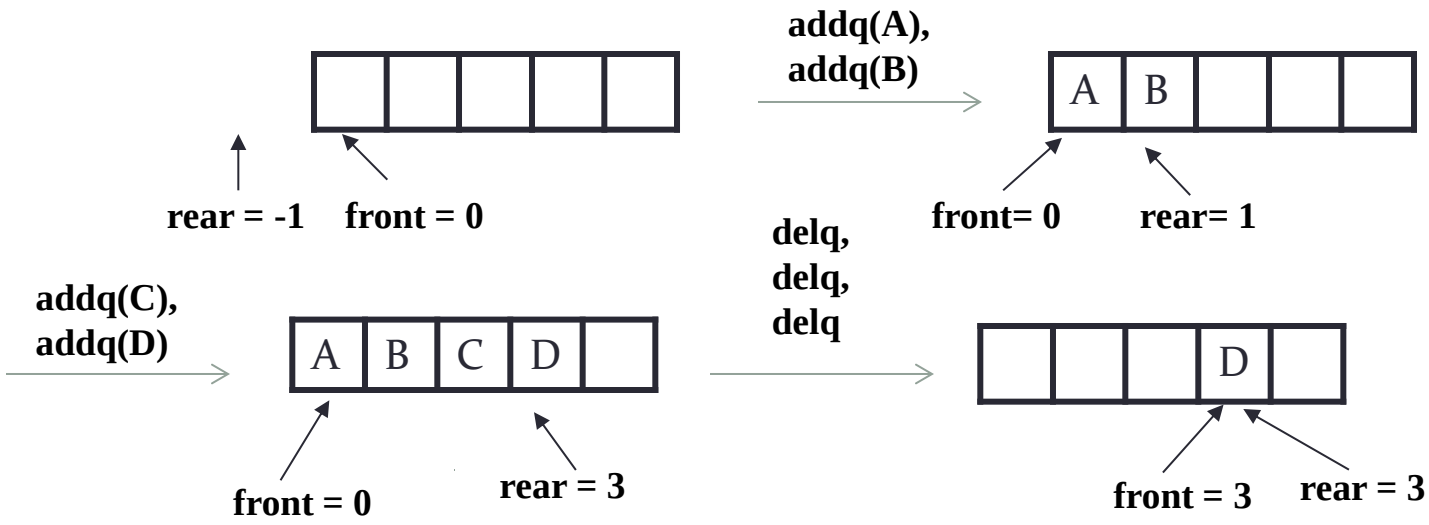
مشکلات پیاده سازی صف با آرایه:



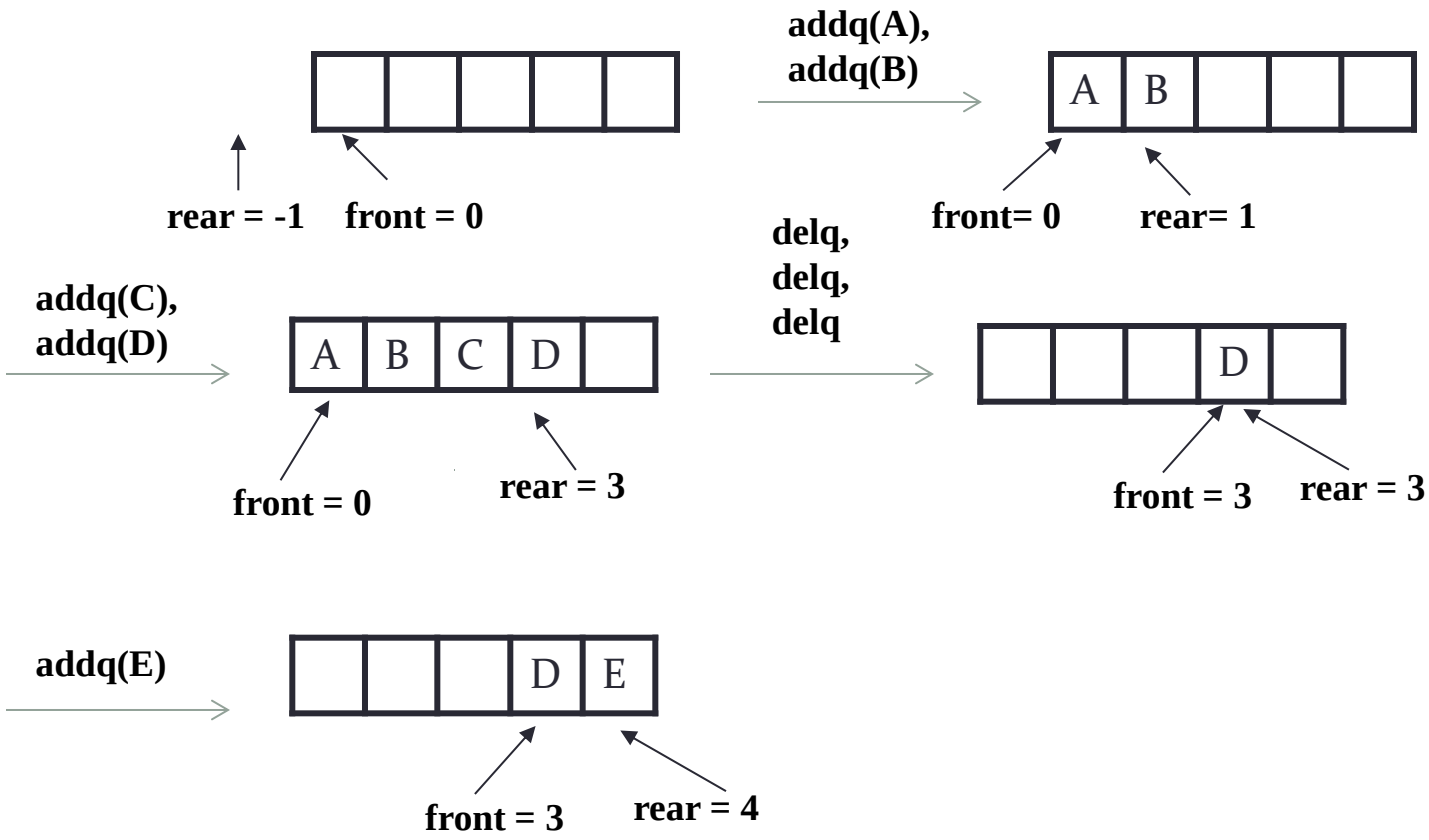
مشکلات پیاده سازی صف با آرایه:



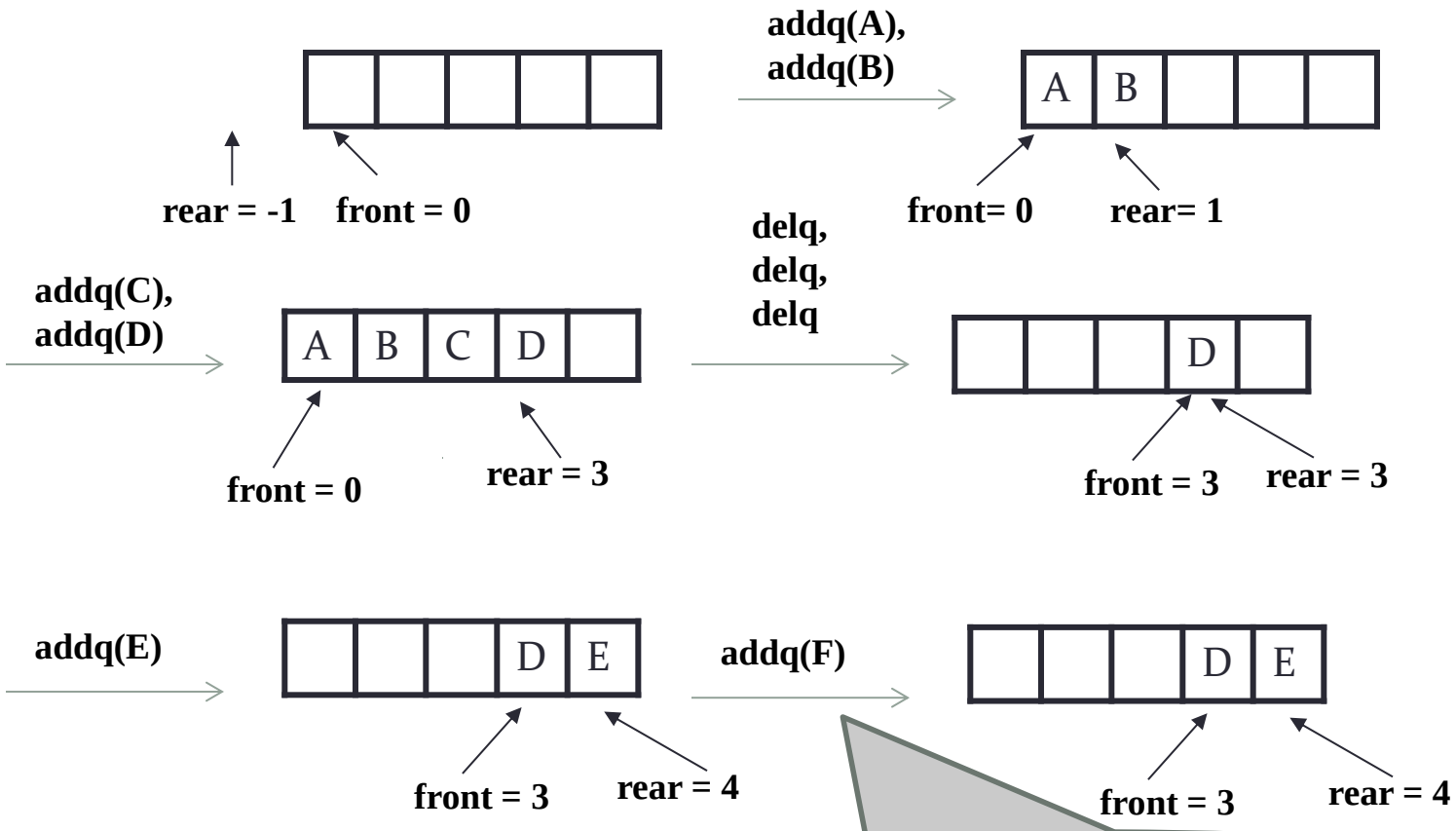
مشکلات پیاده سازی صف با آرایه:



مشکلات پیاده سازی صف با آرایه:



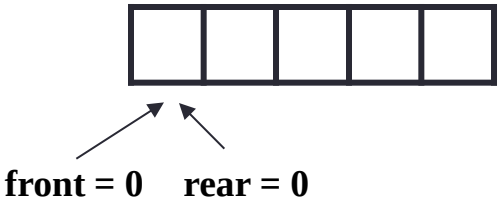
مشکل پیاده سازی صف با آرایه:



اگر عنصر F را با استفاده از تابع addq به صف اضافه نمایید،
overflow = 1 می شود. در صورتیکه صف واقعا پر نیست.

تمرین ۱) زیر برنامه ای به نام printq بنویسید که محتویات صف را چاپ نماید؟

تمرین ۲) زیر برنامه های empty، addq، delq و retrieveq را برای صف زیر بنویسید؟



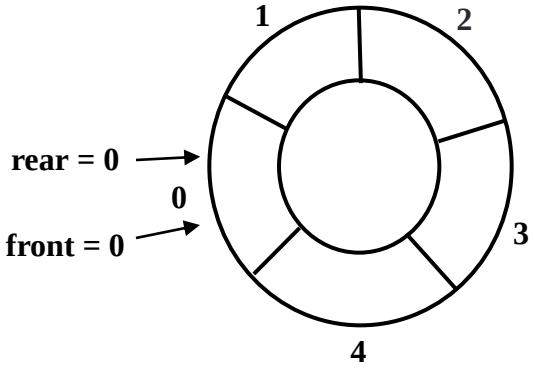
راه حل اول:

عمل حذف از صف طوری انجام شود که پس از حذف عنصری از صف، کلیه عناصر آن به طرف ابتدای آرایه منتقل شوند.

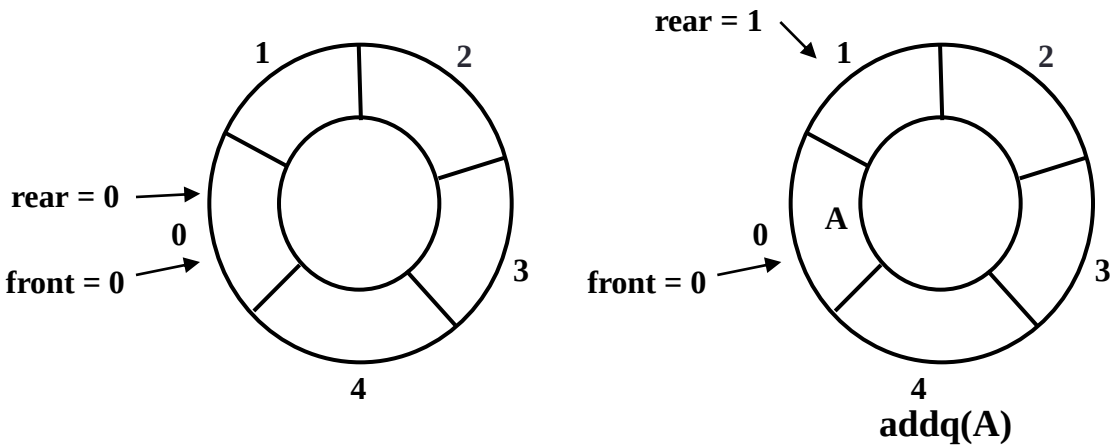
```
for (int i=0; i<rear; i++)
        items[i]=items[i+1];
    rear--;
```

- ✓نیازی به متغیر front نیست.
- ✓صف خالی: rear == -1

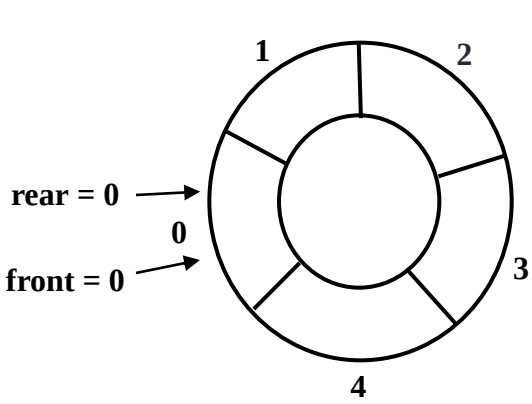
راه حل دوم: صف حلقوی/چرخشی:

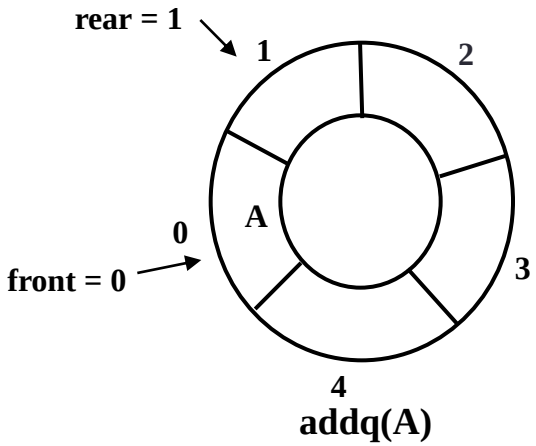


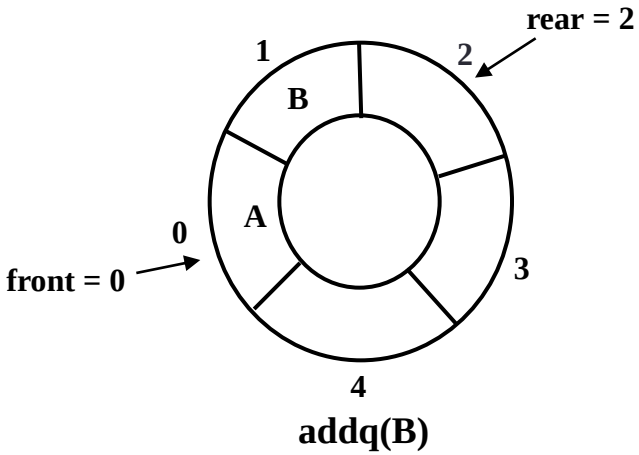
راه حل دوم: صف حلقوی/چرخشی



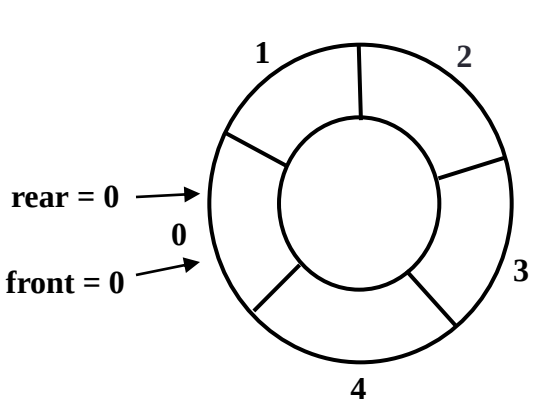
راه حل دوم: صف حلقوی/چرخشی

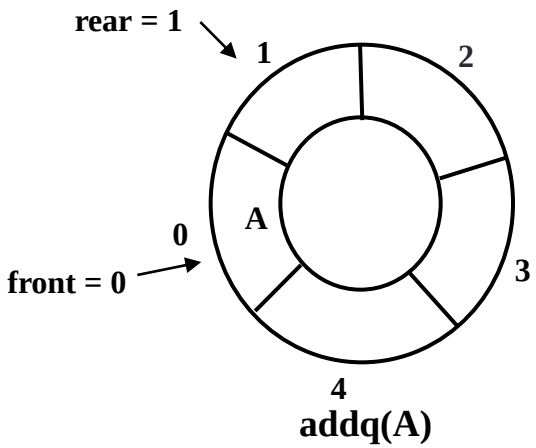




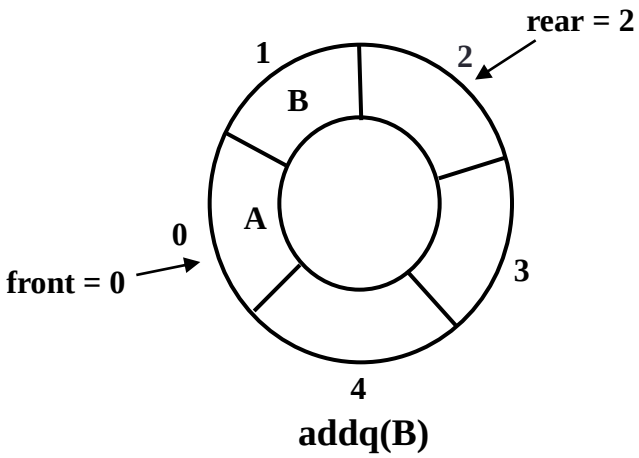


راه حل دوم: صف حلقوی/چرخشی

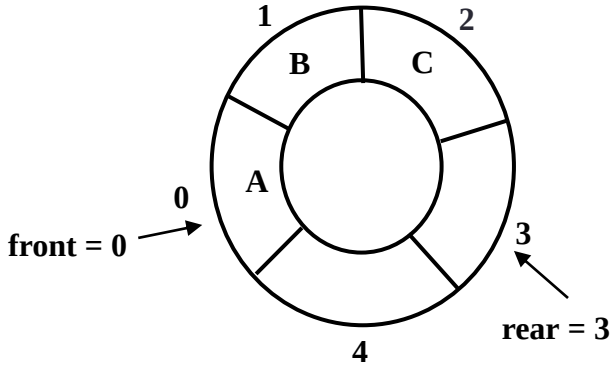




addq(A)

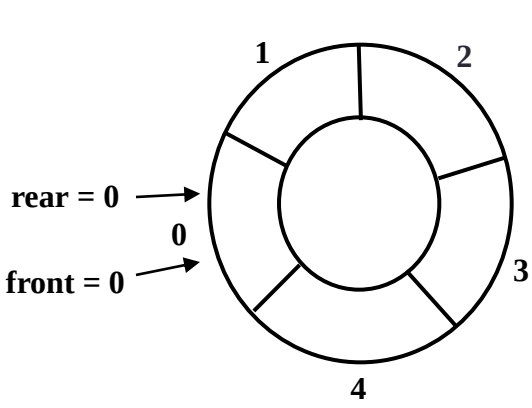


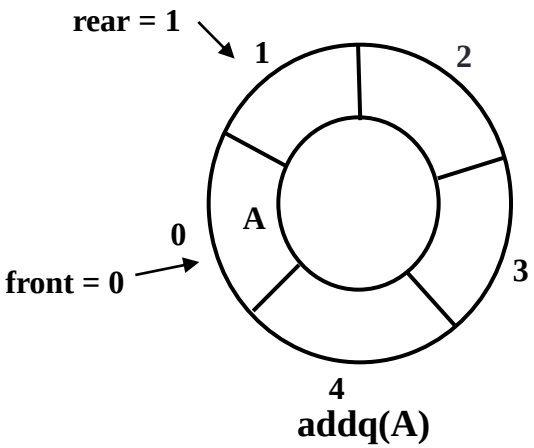
addq(B)



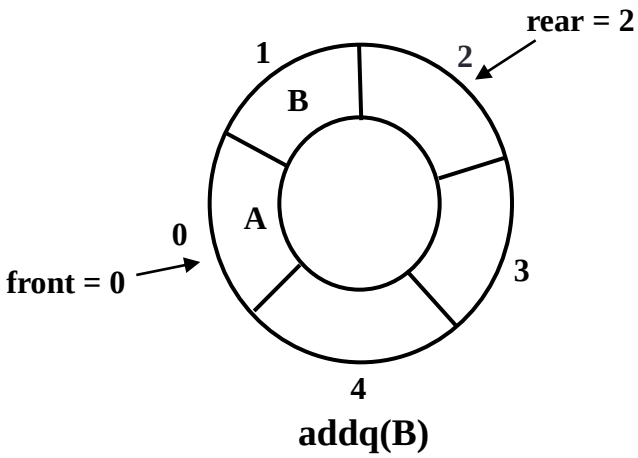
addq(C)

راه حل دوم: صف حلقوی/چرخشی

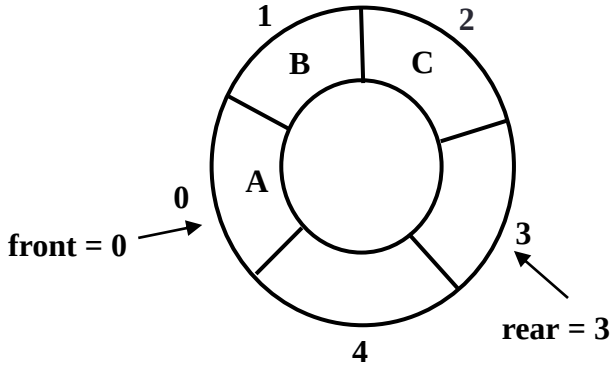




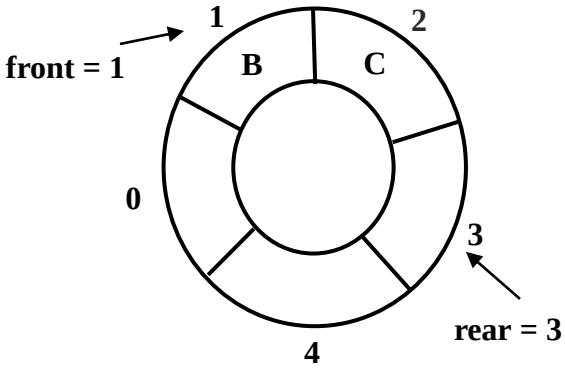
addq(A)



addq(B)

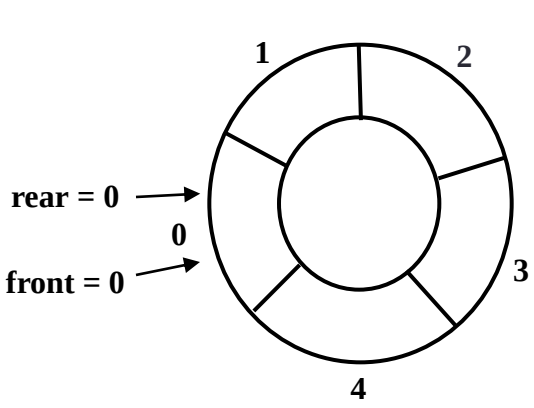


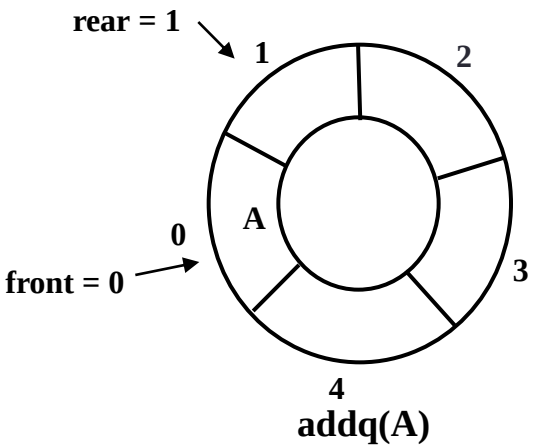
addq(C)



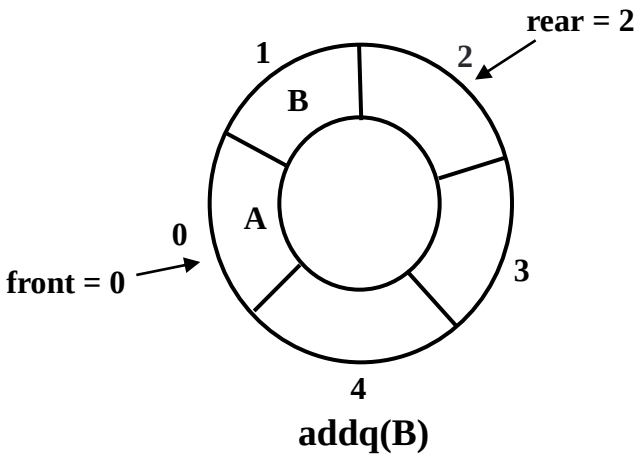
delq()

راه حل دوم: صف حلقوی/چرخشی

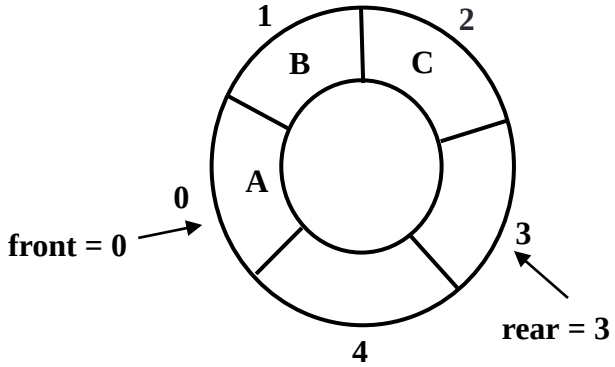




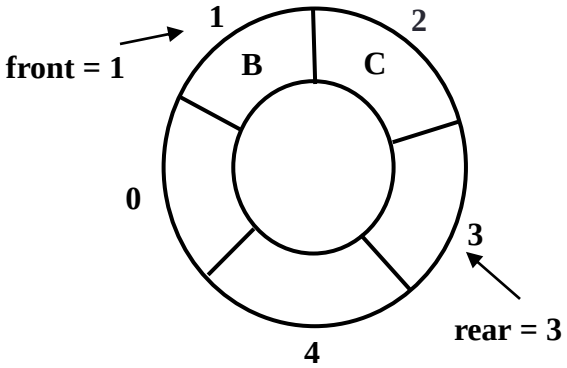
addq(A)



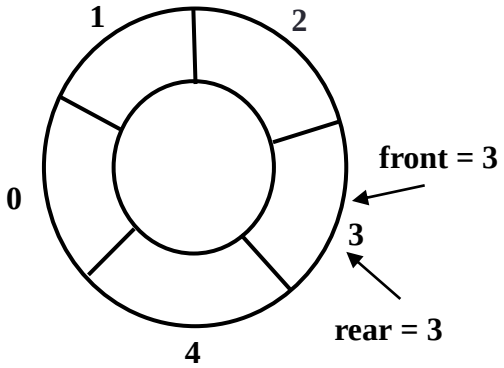
addq(B)



addq(C)



delq()



delq()
delq()

صف حلقوی:

پیاده سازی عمل ایجاد صف حلقوی:

```
queue::queue()
{
  front = rear = 0;
}
```

صف حلقوی:

پیاده سازی عمل ایجاد صف حلقوی:

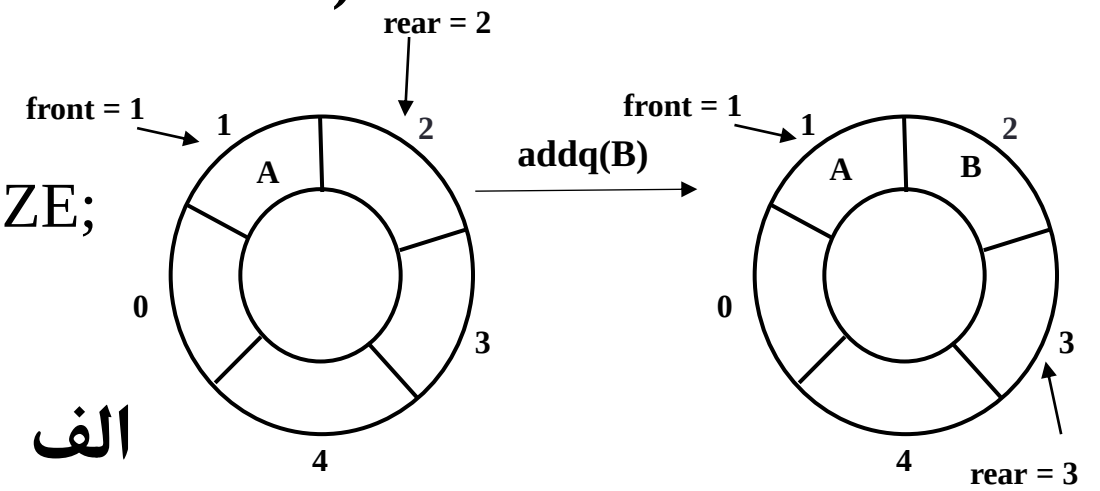
```
queue::queue()
{
  front = rear = 0;
}
```

پیاده سازی عمل بررسی خالی بودن صف حلقوی:

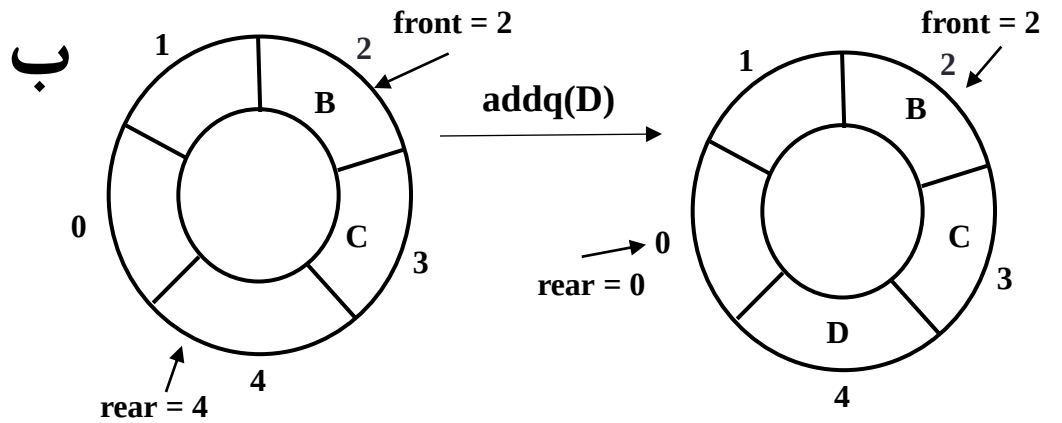
```
int queue::empty()
{
 if (rear == front)
   return 1;
 return 0;
}
```

پیاده سازی عمل افزودن عنصر به صف حلقوی:

```
void queue::addq(int x, int &overflow)
{
   int newrear;
   newrear = (rear +1) % SIZE;
   if (front = = newrear)
         overflow = 1;
    else
        {
          overflow = 0;
          items[rear]= x;
          rear = newrear;
         }
}
```



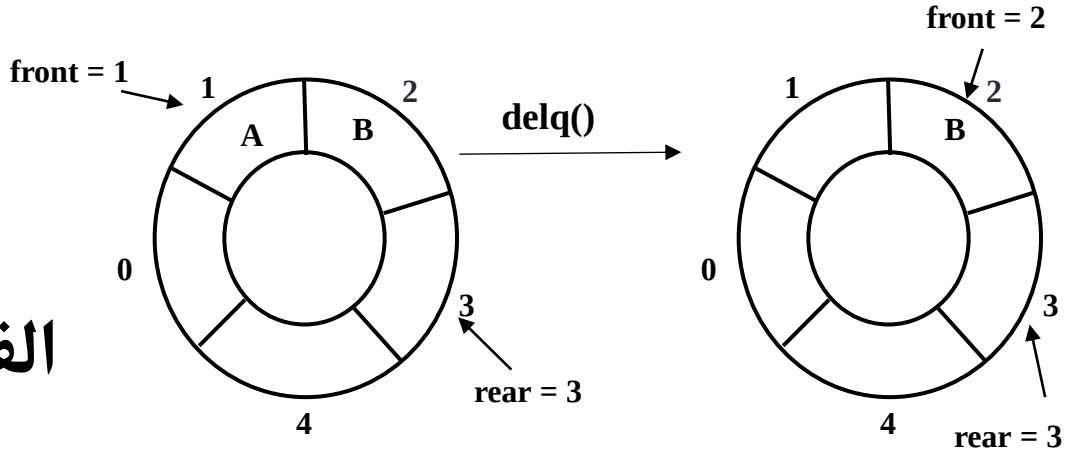
الف



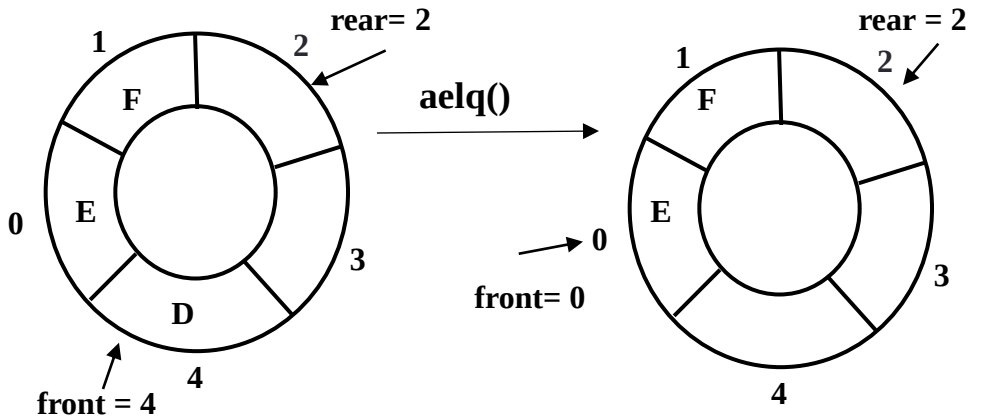
پیاده سازی عمل حذف عنصر از صف حلقوی:

```
void queue:: delq(int &x, int &underflow)
{
  if (empty ())
      underflow=1;
  else
    {
      underflow=0;
      x=items[front];
      front = (front +1) % SIZE;
    }
}
```

الف



ب



پیاده سازی عمل بازیابی عنصر از صف حلقوی:

```
void queue:: retrieveq(int &x, int &underflow)
{
  if (empty ())
      underflow=1;
  else
    {
      underflow=0;
      x=items[front];
    }
}
```

پایان