صف (queue)

صـف لیستی است که عمل افزودن داده‌ها درون آن از یک طرف لیست یا انتهای لیست و عمل حذف داده‌هـا از سـمت دیگر یا ابتدای لیست انجام می‌شود. صف را لیست FIFO (First In First Out) مـی‌نامند. زیرا اولین عنصر ورودی , اولین عنصر خروجی از صف نیز هست. در ساختمان داده صف دو متغیـر front و rear بـه تـرتیب بـرای نشان دادن جلو و انتهای صف بکار می‌روند. صف را می‌توان با استفاده از آرایه‌ها یا لیست‌های پیوندی پیاده سازی کرد.

اگر صف را آرایه‌ای n عضوی از عناصر بدانیم مقادیر front و rear می‌تواند از صفر تا n تغییر کند که برای صف در ابتدا مقادیر اولیه صفر را برای front و rear تعریف می‌کنیم. front = rear = 0

در صورتیکه متغیر front با rear برابر باشد صف خالی است و در صورتیکه rear برابر با n باشد صف پر است.

rear = n ===➔ صف پر است

front = rear ===➔ صف خالی است

دو عمـل اصـلی بـرای صف , حذف کردن داده‌ها از صف و افزودن داده‌ها به صف است که به ترتیب با delqueue و Addqueue نمـایش می‌دهیم. تابع Addqueue(x) به این معنی است که عنصر x بـه انـتهای صـف اضـافه شده است و delqueue نیز مقدار جلوی صف را برداشته و در متغیر x قرار می‌دهد. x = delqueue

پیاده سازی تابع Addqueue و delqueue از صف

queue : Array [1 .. n] of item

برای اضافه کردن

```
void Addqueue (int x)
{
    if (rear = = n)
        C out << " صف پر است " ;
    else
    {
        rear + + ;
        queue[rear] = x ;
    }
}
```

برای حذف کردن

```
int delqueue ( )
{
    if (front = = rear)
    {
        C out << " صف خالی است " ;
        return 0 ;
    }
    else
    {
        front + + ;
        x = queue[front] ;
        return x ;
    }
}
```

مثال : با استفاده از توابع صفحه قبل مقادیر نهایی A و B و C را بدست آورید.

A = 5 B = 10 C = 2 n = 4

Addqueue (A + B)

Addqueue (C)

Addqueue (B × C)

A = delqueue ( )

B = delqueue ( )

Addqueue (B – A)

15	2	20	–13
1	2	3	4

Rear	Front	A	B	C
0	0	5	10	**2**
1	1	**15**	**2**	
2	2			
3				
4				

پس بنابراین داریم : **A = 15** **B = 2** **C = 2**

توابع Addqueue و delqueue به صورتیکه نوشته شد یک صف خطی را پیاده‌سازی می‌کنند. مشکل صف خطی این است که تنها یک بار قابل پر شدن است و در صورتیکه عناصر آن حذف شوند نیز با پیغام « صف پر است » مواجه می‌شوید به همین دلیل صف را بصورت حلقوی تعریف می‌کنیم. در صف حلقوی (دوار) rear و front بعد از رسیدن به آخرین مقدار خود در صورت وجود شرایط لازم مجدداً مقادیر اولیه را می‌توانند بگیرند. صف حلقوی n عضوی را بصورت آرایه صفر تا n – 1 تعریف می‌کنیم.

queue : Array [0 .. n – 1] of item

در این حالت وقتی rear = n – 1 عنصر بعدی در queue[0] قرار می‌گیرد. در صف حلقوی front = rear به معنای خالی بودن صف است ولی شرط پر بودن صف بدین ترتیب تغییر می‌یابد.

front = (rear + 1) mod n ===➔ شرط پر بودن

front = rear ===➔ صف خالی است

برای اضافه کردن به صف حلقوی , rear یکی اضافه می‌شود و در صورتیکه rear = n – 1 باید صفر بشود. بدین منظور rear را با رابطه زیر در هر شرایطی مقداردهی می‌کنند.

rear = (rear + 1) mod n

این مسئله برای front نیز برقرار است.

front = (front + 1) mod n

برای حذف کردن

```
int delqueue ( )
{
    if (front = = rear)
    {
        C out << " صف خالی است " ;
        return 0 ;
    }
    else
    {
        front = (front + 1) mod n
        x = queue[front] ;
    }
}
```

برای اضافه کردن

```
void Addqueue (int x)
{
    rear = (rear + 1) mod n
    if (front = = rear)
        C out << " صف پر است " ;
    else
        queue[rear] = x ;
}
```

مثال :

Addqueue [50]	r = 1	0	1	2	3
	f = 0		50		
Addqueue [20]	r = 2	0	1	2	3
	f = 0		50	20	
Addqueue [30]	r = 3	0	1	2	3
	f = 0		50	20	30
delqueue ( )	r = 3	0	1	2	3
	f = 1		50	20	30
Addqueue [10]	r = 0	0	1	2	3
	f = 1	10	~~50~~	20	30