



МАССИВЫ

Циклические сдвиги,
сжатие

Алгоритм циклического сдвига на k позиций.

I способ

1. определить сколько раз необходимо произвести одноэлементный сдвиг

$$k := k \% n;$$

2. k раз применить одноэлементный сдвиг
Алгоритм одноэлементного сдвига.

- 1) Запомнить в дополнительной ячейке первый (или последний) элемент массива
- 2) Сдвинуть все элементы влево (вправо)
- 3) На последнее (первое) место записать тот, который запоминали.

Сдвиг вправо и влево

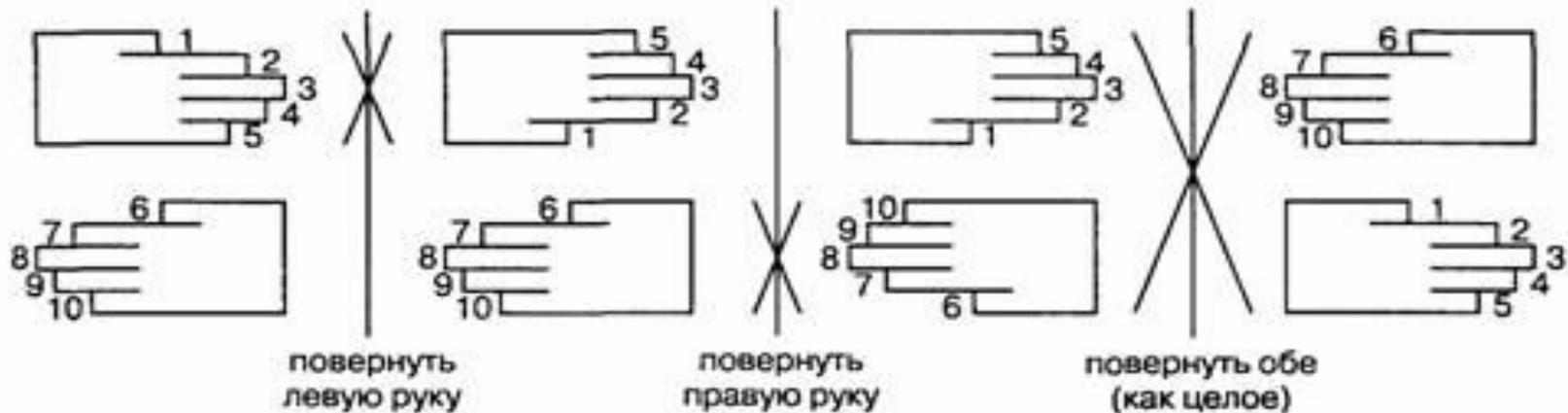
```
n=int(input())
a=[5]*n
for i in range(n):
    a[i]=int(input())
print(a)
k=int(input())
k=k%n
for i in range(k):
    t=a[0]
    for j in range(n-1):
        a[j]=a[j+1]
    a[n-1]=t
print(a)
```

II способ

1. Скопировать первые k элементов массива во временный массив
2. Сдвинуть оставшиеся $n-k$ элементов влево на k позиций
3. Скопировать данные из временного массива обратно в основной массив на последние k позиций

III способ

1. отобразить элементы массива $(0, k-1)$
2. отобразить элементы массива $(k, n-1)$
3. отобразить элементы массива $(0, n-1)$



j-сколько раз произвести обмен, **left** - левая граница отображения, **right** - правая граница отображения, **Dlina** - длина отображаемой части массива

$j=1$ $left=0$ $right=k-1$ $dlna=right-left+1$

(***) **while** $j \leq dlna // 2$:

$temp=a[left]$

$a[left]=a[right]$

$a[right]=temp$

$left+=1$

$right-=1$

$j+=1$

$j=1$ $left=k$ $right=n-1$ $dlna=right-left+1$

(***) {повторить цикл}

$j=1$ $left=0$ $right=n-1$ $dlna=right-left+1$

(***) {повторить цикл}

Сжатие массива.

Удаление каждого **k**-го элемента:

i – индекс активного элемента

l - индекс просматриваемого элемента

kol – количество элементов после всех удалений.

```
i=k-1; l=k-1;
```

```
while l<=n-1:
```

```
    if (l+1) % k==0 :
```

```
        l+=1
```

```
    if l<=n-1 :
```

```
        a[i]=a[l];
```

```
    i+=1
```

```
    l+=1
```

```
kol=n-n // k
```