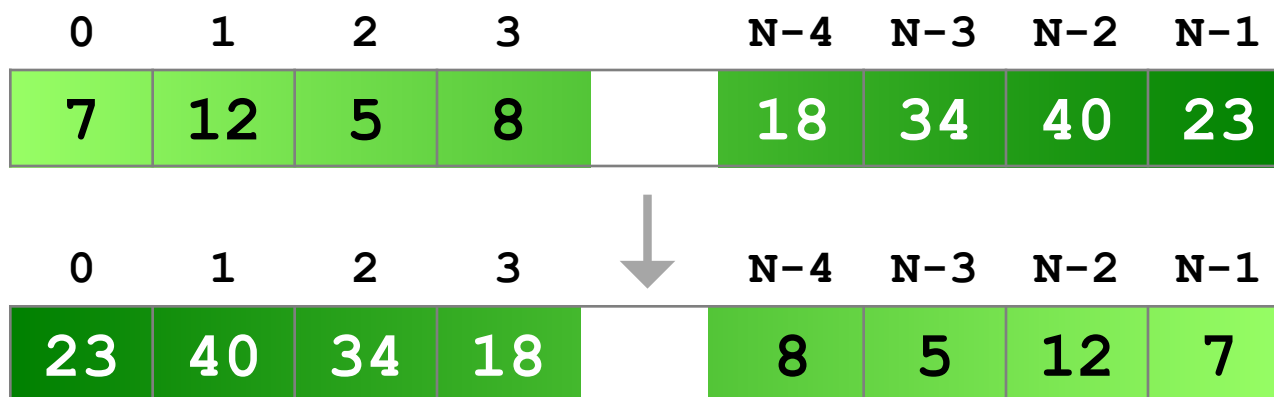


Реверс массива



«Простое» решение:

остановиться на середине!

```
for i in range(N//2):
    поменять местами A[i] и A[N-1-i]
```



Что плохо?

Реверс массива

```
for i in range(N//2):  
    c = A[i]  
    A[i] = A[N-1-i]  
    A[N-1-i] = c
```

Варианты в стиле Python:

```
for i in range(N//2):  
    A[i], A[N-i-1] = A[N-i-1], A[i]
```

```
A.reverse()
```

Циклический сдвиг элементов

0	1	2	3		N-4	N-3	N-2	N-1
7	12	5	8		18	34	40	23
↓								
0	1	2	3		N-4	N-3	N-2	N-1
12	5	8	15		34	40	23	7

«Простое» решение:

```
for i in range(N-1):
    A[i] = A[i+1]
```

?

Почему не до N?

?

Что плохо?

Срезы в Python

0	1	2	3		N-4	N-3	N-2	N-1
7	12	5	8		18	34	40	23

! Последний элемент не входит в срез!

$A[1:3] \rightarrow [12, 5]$

$A[2:3] \rightarrow [5]$

$A[:3] \rightarrow A[0:3] \rightarrow [7, 12, 5]$

с начала

$A[3:N-2] \rightarrow [8, \dots, 18, 34]$

$A[3:] \rightarrow A[3:N] \rightarrow [8, \dots, 18, 34, 40, 23]$

до конца

копия массива

$A[:] \rightarrow [7, 12, 5, 8, \dots, 18, 34, 40, 23]$

Срезы в Python – отрицательные индексы

0	1	2	3		N-4	N-3	N-2	N-1
7	12	5	8		18	34	40	23
-N	-N+1	-N+2	-N+3		-4	-3	-2	-1

$A[1:-1]$ → [12, 5, 8, ..., 18, 34, 40]
 $A[1:N-1]$

$A[-4:-2]$ → [18, 34]
 $A[N-4:N-2]$

Срезы в Python – шаг

0	1	2	3	4	5	6	7	8
7	12	5	8	76	18	34	40	23

шаг

`A[1:6:2]` → [12, 8, 18]

`A[::3]` → [7, 8, 34]

`A[8:2:-2]` → [23, 34, 76]

`A[::-1]` → [23, 40, 34, 18, 76, 8, 5, 12, 7]

реверс!

`A.reverse()`

Задачи

«А»: Заполнить массив случайными числами и выполнить циклический сдвиг элементов массива вправо на 1 элемент.

Пример:

Массив :

1 2 3 4 5 6

Результат:

6 1 2 3 4 5

«В»: Массив имеет четное число элементов. Заполнить массив случайными числами и выполнить реверс отдельно в первой половине и второй половине.

Пример:

Массив :

1 2 3 4 5 6

Результат:

3 2 1 6 5 4

Отбор нужных элементов

Задача. Отобрать элементы массива **A**, удовлетворяющие некоторому условию, в массив **B**.

Простое решение:

```
B = []  
сделать для i от 0 до N-1  
    если условие выполняется для A[i] то  
        добавить A[i] к массиву B
```

```
B = []  
for x in A:  
    if x % 2 == 0:  
        B.append(x)
```



Какие элементы выбираем?

добавить **x** в конец
массива **B**

Отбор нужных элементов

Задача. Отобрать элементы массива **A**,
удовлетворяющие некоторому условию, в массив **B**.

Решение в стиле Python:

перебрать все
элементы A

```
B = [ x for x in A  
      if x % 2 == 0 ]
```

если **x** – чётное
число

Особенности работы со списками

```
A = [1, 2, 3]
```

```
B = A
```

A → [1, 2, 3]
B → [1, 2, 3]



```
A[0] = 0
```

A → [0, 2, 3]
B → [0, 2, 3]

```
A = [1, 2, 3]
```

```
B = A[:]
```

копия массива A

A → [1, 2, 3]



A → [0, 2, 3]

B → [1, 2, 3]

```
A[0] = 0
```

B → [1, 2, 3]

Копирование списков

«Поверхностное» копирование:

```
import copy
A = [1, 2, 3]
B = copy.copy(A)
```

A → [1, 2, 3]
B → [4, 5, 6]

```
A = [1, 2, 3]
B = [4, 5, 6]
C = [A, B]
D = copy.copy(C)
C[0][0] = 0
```

C → [A, B] A → [0, 2, 3]
D → [A, B] B → [4, 5, 6]

! Влияет на C и D!

«Глубокое» копирование:

```
D = copy.deepcopy(C)
```

C → [A, B] A → [1, 2, 3]
D → [·, ·] B → [4, 5, 6]
 [1, 2, 3]
 [4, 5, 6]

Задачи

«А»: Заполнить массив случайными числами в интервале $[-10, 10]$ и отобразить в другой массив все чётные отрицательные числа.

Пример:

Массив А:

-5 6 7 -4 -6 8 -8

Массив В:

-4 -6 -8

«В»: Заполнить массив случайными числами в интервале $[0, 100]$ и отобразить в другой массив все простые числа. Используйте логическую функцию, которая определяет, является ли переданное ей число простым.

Пример:

Массив А:

12 13 85 96 47

Массив В:

13 47