

Примеры решения задач

1. Пользователь вводит число. Вывести «Да», если число – степень двойки, иначе вывести «Нет». Задача решается при помощи рекурсии.

```
a = int(input())
p = lambda x: 1 if x == 1 else 0 if 1 < x < 2 else p(x / 2)
res = lambda x: 'Да' if p(x) == 1 else 'Нет'
print(res(a))
```

2. Вычислить сумму цифр введенного с клавиатуры целого числа.

```
print(sum(map(int, list(input()))))
```

3. Получить список результатов вычисления $a^2 + b - 1$ при заданных значениях a и b .

```
a = [1, 2, 3]
b = [-1, 5, 2]
print(list(map(lambda x, y: x**2 + y - 1, a, b)))
[-1, 8, 10]
```

4. Вывести первые 10 элементов ряда чисел Фибоначчи.

```
fib = lambda x: 0 if x == 0 else 1 if 0 < x <= 2 else fib(x - 1) + fib(x - 2)
print(list(map(fib, range(10))))
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

5. Вычислить произведение элементов списка.

```
from functools import reduce
a = [2, 4, 6, 8]
print(reduce(lambda x, y: x * y, a))
384
```

Порядок выполнения: $((2 * 4) * 6) * 8$

6. Вычислить сумму квадратов чисел из списка.

```
from functools import reduce
sr = lambda x: x**2
sm = lambda x, y: x + y
L = [1, 2, 3, 4]
print(reduce(sm, map(sr, L)))
30
```

7. Получить одномерный список всех элементов из заданного двумерного списка без повторов.

```
from functools import reduce
L = [[2, 4, 6, 8], [1, 3, 5], [1, 2, 3, 4, 7, 12]]
print(list(reduce(set.union, map(set, L))))
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 12]
```

8. Получить список словарей с данными о людях, живущих в Новосибирске.

```
humans = [ {'name': 'Глеб', 'age': 20, 'city': 'Новосибирск'},  
           {'name': 'Марк', 'age': 31, 'city': 'Кемерово'},  
           {'name': 'Анна', 'age': 17, 'city': 'Новосибирск'} ]  
print(list(filter(lambda x: x['city'] == 'Новосибирск', humans)))
```

```
[{'city': 'Новосибирск', 'name': 'Глеб', 'age': 20},  
 {'city': 'Новосибирск', 'name': 'Анна', 'age': 17}]
```