

Примеры решения задач

1. Написать функцию вычисления суммы цифр натурального числа. Привести пример использования.

```
def summa(a):
    s = 0
    while a > 0:
        s = s + a % 10
        a = a // 10
    return s

b = int(input())
print(summa(b))
```

2. Написать функцию проверки существования треугольника по длинам трех сторон.

```
def istriangle(a, b, c):
    if ((a + b > c) and (a + c > b) and (a + c > b)):
        return 'Yes'
    else:
        return 'No'

print('Введите 3 числа')
a1 = int(input('a1='))
a2 = int(input('a2='))
a3 = int(input('a3='))
print(istriangle(a1, a2, a3))
```

3. Последовательность Фибоначчи задана следующим образом: $F_0=0$, $F_1=1$, при $n>1$ число Фибоначчи с номером n вычисляется как $F_n=F_{n-1}+F_{n-2}$. Написать рекурсивную функцию для получения n -го числа Фибоначчи.

```
def fb(n):
    if n <= 1:
        return 1
    else:
        return fb(n - 1) + fb(n - 2)

print(fb(5))
```

8

Функция расчета чисел Фибоначчи может быть улучшена при помощи словаря. Если нам придется вычислять несколько чисел Фибоначчи в процессе исполнения программы с нуля, это займет много времени. Это время можно сэкономить, если сохранять все вычисленные значения, включая промежуточные, в словаре следующим образом:

```
alreadyknown = {0: 0, 1: 1}
def fb(n):
    if n not in alreadyknown:
        new_value = fb(n - 1) + fb(n - 2)
        alreadyknown[n] = new_value
    return alreadyknown[n]
```

Таким образом, если нам необходимо вычислить:

fb(5)

fb(12)

fb(8)

Сначала будут вычислены первые 5 чисел и выдан ответ на первый запрос, затем вычислены только значения для 6–12 чисел и выдан ответ, а ответ на третий запрос будет выдан сразу, поскольку 8 число уже было вычислено и хранится в словаре.

Полный код программы:

```
alreadyknown = {0: 0, 1: 1}
def fb(n):
    if n not in alreadyknown:
        new_value = fb(n - 1) + fb(n - 2)
        alreadyknown[n] = new_value
    return alreadyknown[n]

print(fb(5))
print(fb(12))
print(fb(8))
```