

Задание №16

Вычисление рекурсивных выражений

5 мин

1 балл

Повышенный уровень

Программирование

Кодификатор

3.7

Элементы содержания

Рекурсия. Рекурсивные процедуры и функции. Использование стека для организации рекурсивных вызовов

1.8

Предметные требования

Владение теоретическим аппаратом, позволяющим осуществлять представление заданного натурального числа в различных системах счисления; выполнять преобразования логических выражений, используя законы алгебры логики; определять кратчайший путь во взвешенном графе и количество путей между вершинами ориентированного ациклического графа

Рекурсия

Решать сложную задачу - **сложно**
Легче сделать из этой задачи простую, такую же по структуре, и решить уже её



Рекурсия включает в себя два основных элемента:

- 1. Базовый (крайний) случай:** Это условие, при котором функция перестает вызывать саму себя и возвращает некоторый конкретный результат. Без базового случая рекурсивная функция может вызываться бесконечно.
- 2. Шаг рекурсии:** Это часть функции, где происходит вызов функцией самой себя с измененными параметрами. Этот шаг должен приближать к достижению базового случая.

Вычисление факториала

Линейный алгоритм

$$n! = 1 * 2 * 3 * \dots * n$$

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

 main.py

```
1 def f(n):
2     factorial = 1
3     for x in range(2, n + 1):
4         factorial *= x
5     return factorial
```

Рекурсивный алгоритм

$$1! = 1$$

$$n! = n * (n - 1)!$$

$$\begin{aligned} 5! &= 5 * 4! = 5 * 4 * 3! = 5 * 4 * 3 * 2! = \\ &= 5 * 4 * 3 * 2 * 1! = 5 * 4 * 3 * 2 * 1 = 120 \end{aligned}$$

 main.py

```
1 def f(n):
2     if n == 1:
3         return 1
4     else:
5         return n * f(n - 1)
```