

Необходимо создать 1 файл-модуль с именем формата xxx.py, где xxx - фамилия латинскими буквами.

В нем последовательно объявляются функции (taskN) с решением каждой из задач. Вызовы функций выполняются произвольно.

# пример заполнения файла

```
def task1(a, b, n):  
    print 'Something useful'  
  
def task2(n):  
    assert type(n) == int, 'Not int!'  
    print 'haha'
```

```
task2(620)
```

По окончании работы - необходимо загрузить файл в систему LMS в соответствующий проект.

Оценивается:

- 1) количество выполненных задач
- 2) качество (решают ли они поставленную задачу и в какой мере)
- 3) наличие проверок на типы, условия и т.п.
- 4) наличие комментариев приветствуется

## Общие

1. Вывести на экран все натуральные числа из диапазона от A до B в записи которых цифра 7 встречается N раз. При отсутствии чисел с указанными свойствами выдать на экран сообщение "Требуемых чисел нет". Границы диапазона A и B и значение N вывести с клавиатуры. *(Рекомендуется использовать методы работы со строками)*
2. Ввести натуральное трехзначное число. Вычислить и вывести на экран число, полученное путем "переворота" (123 -> 321). Не использовать методы для работы со строками.
3. Ввести натуральное четырехзначное десятичное число, сформировать и вывести на экран признак "счастливого числа" (сумма первых двух цифр равна сумме последних двух). Не использовать методы для работы со строками.
4. Реализовать метод вычисления факториала.

5. Вычислить значение функции  $Y=F(X)$ , заданной графиком

$$Y = \begin{cases} 0.5 & \text{при } X \leq -0.5 \\ X+1 & \text{при } -0.5 < X \leq 0 \\ X^2 X - 1 & \text{при } 0 < X \leq 1 \\ X-1 & \text{при } X > 1 \end{cases}$$

6.  $G = \begin{cases} \sin(\pi/2), & \text{если } X \leq 0.5 \end{cases}$

$$G = \begin{cases} \sin((X-1)\pi/2), & \text{если } X > 0.5 \end{cases}$$

## Циклы

7. Написать метод для вычисления приближенного значения **кубического** корня вещественного числа  $x$  с помощью итерационного метода:

$$a_{i+1} = a_i + \frac{(x - a_i^3)}{3 * a_i^2}, \text{ где } i = 0, 1, \dots, a_0 = x$$

Итерации продолжать пока  $a_{i+1} \neq a_i$  и очередное приращение  $\frac{(x - a_i^3)}{3 * a_i^2}$  по модулю превышает  $x * 1e - 10$ . ( $10.0^{**} - 10$ )

Метод должен возвращать количество итераций  $i$ , вычисленное значение корня

$$a_{i+1} \text{ и в качестве оценки точности – величину } \frac{(x - a_i^3)}{3 * a_i^2}$$

8. Сократить введенную обыкновенную дробь. Дробь вводится с клавиатуры в виде числителя и знаменателя. Как вспомогательную функцию определить и использовать метод для вычисления наибольшего общего делителя двух целых чисел.
9. Найти и распечатать все натуральные трехзначные числа, равные сумме кубов своих цифр.
10. Даны натуральные числа  $n$  и  $k$ . Определить  $k$ -ю справа цифру числа  $n$ .
11. Вычислить  $k$ -ое число Фибоначчи.

## Массивы

12. Ввести натуральное число  $N$ . Одномерный массив размером  $N$  заполнить случайными числами. Вывести на экран количество простых чисел в массиве (Для удобства проверки дополнительно вывести сгенерированный массив).
13. Дан целочисленный массив  $A$ . Найти в нем два наименьших элемента.

14. Сгенерировать (и вывести) двумерную матрицу размером  $N \times N$  ( $N < 20$ ). Найти сумму ее элементов, находящихся на диагонали, и сумму элементов на диагонали, "ортогональной" главной.
15. Сформировать целочисленный массив  $A(75)$ , элементами которого являются случайные числа из диапазона  $[-5, 20]$ . Найти среди его элементов два, разность которых имеет наибольшее значение.
16. Найти наибольший общий делитель (НОД) двух введенных натуральных чисел, используя алгоритм Евклида. Алгоритм Евклида: вычитаем из большего числа меньшее до тех пор, пока они не сравняются; полученное в результате число и есть НОД.