# Задания к самостоятельной работе

1. Напишите функцию integerPower(base,exponent), которая возвращает значение baseexponent . Например, integerPower(3,4)=3\*3\*3\*3. Предположим, что exponent является положительным ненулевым целым, а base целым. Для управления вычислением функция integerPower(base,exponent) должна применять цикл for. Не используйте никаких функций математической библиотеки.

2. Напишите функцию multiple для двух целых, которая определяет, кратно ли второе число первому. Функция должна получать два целых аргумента и возвращать 1, если второе число кратно первому и 0 в противном случае. Используйте эту функцию в программе, которая вводит серию пар целых чисел.

3. Напишите функцию, которая выводит у левой границы экрана сплошной квадрат из символов заданного символа (например, \* ), сторона которого определяется целым параметром side. Например, если side равно 4, функция выведет следующее изображение

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

4. Даны действительные числа s и t.

Используйте функции и вычислите

 , где 

5. Даны действительные числа a,b,c. Использовать функции и получить



6. Напишите программу, которая вводит последовательность целых чисел и передает их по одному функции even, использующей операцию вычисления остатка для определения четности числа. Функция должна принимать целый аргумент и возвращать 1, если аргумент — четное число, и 0 — в противном случае.

7. Говорят, что целое число является совершенным,если его сомножители, включая 1 (но не само число) в сумме дают это число. Например, 6 — это совершенное число, так как 6=1 + 2 + 3. Напишите функцию perfect, которая определяет, является ли параметр number совершенным числом. Используйте эту функцию в программе, которая определяет и печатает все совершенные числа в диапазоне от 1 до 1000. Напечатайте сомножители каждого совер­шенного числа, чтобы убедиться, что число действительно совер­шенное.

8. Говорят, что целое число является простым числом,если оно, де­лится только на 1 и на само себя. Например, 2, 3, 5 — простые числа, а 4, 6, 8 — нет.

1. Напишите функцию, определяющую, является ли число простым или нет.
2. Используйте эту функцию в программе, которая определяет и печатает все простые числа, лежащие в диапазоне от 1 до 10000.
3. Вначале вы могли бы подумать, что верхней границей, до которой вы должны проводить проверку, чтобы увидеть, является ли число n простым, является n/2, но в действительности вам нужно прове­рить количество чисел, равное корню квадратному из n. Почему? Перепишите программу и запустите ее для обоих способов. Оцените  
   улучшение производительности.

9. Напишите программу, моделирующую бросание монеты. Для каж­дого броска монеты программа должна печатать **Орел** или **Решка.** Промоделируйте с помощью этой программы бросание 100 раз и подсчитайте, сколько раз появилась каждая сторона монеты. Напечатайте результаты. Программа должна вызывать отдельную функ­цию **flip,** которая не принимает никаких аргументов и возвращает 0 для **Орла** и 1 для **Решки.**

Замечание: если программа действи­тельно моделирует бросание монеты, каждая сторона монеты должна появляться примерно в половине случаев.