**Задание 1**

Реализовать приложение, вычисляющее для последовательности чисел, представляющих собой величины углов, следующие тригонометрические функции:

a.   сумма синусов/косинусов значений,

b.   разность синусов/косинусов значений,

c.   произведение синусов/косинусов введенных значений.

Единица измерения углов должна указываться как параметр командной строки программы (градусы или радианы). Для представления единиц измерения используйте константы с типом int. Для хранения значений, необходимо использовать массив с типом double. Объявления методов должны выглядеть следующим образом:

/\*\*

@param angles – массив значений углов;

@param type -единица измерения, имеющее значение од-

ной из констант

@return сумму синусов значений углов;

\*/

public static double getSumSinuses(double[] angles, int type);

См. класс Math

**Задание 2**

Создать метод вычисляющий факториал. Вычислить факториал достаточно большое количество раз (миллион или больше), сравните эффективность рекурсивного и итерационного алгоритмов. Во сколько раз будет отличаться время выполнения и как это отношение будет зависеть от числа, факториал которого рассчитывается?  Программа должна вывести на экран результат анализа.

**Задание 3**

Создать метод, который будет сортировать по убыванию  одномерный массив случайных чисел из n элементов, переданный в качестве аргумента в этот метод. Сортировка должна быть устойчивой и иметь естественное поведение. (Не использовать Arrays.sort). Программа должна вывести на экран исходный и отсортированный массив.

**Задание 4**

В массиве хранится n явно заданных текстовых строк. Создать программу, которая отсортирует и выведет на экран строки в алфавитном порядке. Указание: прежде всего надо создать метод, устанавливающий отношения порядка для двух строк, переданных методу через аргументы. Программа должна вывести строки в начальном и отсортированном порядке.(2 балла)

Дополнительно 1 балл за генерацию уникальных строк реализованных в виде метода.

**Задание 5**

Королю нужно убить дракона, но в его казне мало средств для покупки армии. Нужно создать программу используя методы, которая поможет рассчитать минимальное количество копейщиков, которое необходимо, чтобы убить дракона. Данные здоровья и атаки дракона и одного копейщика вводятся с клавиатуры.

  Защита, меткость и т. п. не учитывать. Копейщики наносят удар первыми(общий нанесенный урон – это сумма атак всех живых копейщиков). Если атака дракона превышает значение жизни копейщика (например у копейщика жизни 10, а атака 15),  то умирает несколько копейщиков, а оставшийся урон идет одному из копейщиков.

Например, жизнь дракона 500, атака 55, жизнь одного копейщика 10, атака 10, а количество копейщиков при данных условиях 20.

Лог боя для данного примера должен выглядеть так:

Копейщики атакуют (урон 200) – у дракона осталось 300 жизней

Дракон атакует – осталось 15 копейщиков, один из которых ранен (осталось 5 жизней)

Копейщики атакуют – у дракона осталось 150 жизней

Дракон атакует – осталось 9 копейщиков

Копейщики атакуют – у дракона осталось 60 жизней

Дракон атакует – осталось 4 копейщика , один из которых ранен (осталось 5 жизней)

 Копейщики атакуют – у дракона осталось 20 жизней

Дракон атакует  и побеждает

Задание 6

Напишите рекурсивную функцию, проверяющую правильность расстановки скобок в строке введенной с клавиатуры. При правильной расстановке выполняются условия:

a.   количество открывающих и закрывающих скобок равно.

b.   внутри любой пары открывающая – соответствующая закрывающая скобка,

скобки расставлены правильно.

В строке могут присутствовать скобки как круглые, так и квадратные скобки (и др. символы). Каждой открывающей скобке соответствует закрывающая того же типа (круглой – круглая, квадратной- квадратная). Напишите рекурсивную функцию, проверяющую правильность расстановки скобок в строке.

Пример неправильной расстановки: ( [ ) ].

Пример правильных входных данных (a[b](f[(g)(g)]))

Программа должна вывести результат в виде строки, примеры

•  Правильная строка

•  Ошибка отсутствие (

•  Ошибка отсутствие [

За выполнение каждого задания 2 балла