

Все задания выполняются на одном из предложенных языков: C++, Java, Python.

Вам нужно прислать ссылку на ваш публичный репозиторий на github.com.

Имя репозитория должно содержать имя языка: `cpp`, `java`, `python`.

В вашем репозитории должны быть 4 каталога:

`task1`, `task2`, `task3`, `task4`

В каждой из них каталог 'SRC' с исходными файлами (`*.cpp`, `*.hpp`, `*.java`), если C++ или Java, и исполняемый файл (`task#.exe` для C++, `task#.jar` для Java, `task#.py` для Python).

Имена файлов с входными данными передаются через аргументы командной строки.

Результат выполнения программы должен выводиться в консоль.

В качестве разделителя нужно использовать символ перехода на новую строку.

Если порядок вывода результатов будет нарушен, то решение не будет зачтено.

Обратите внимание, что тестовые файлы будут наши. Если вы захардкодите путь к своему файлу, то это будет ошибка.

Примечание для C++.

Ваши приложения будут запускаться на Windows 10 Корпоративная.

В случае, если ваш исполняемый файл не запустится, то он будет заново скомпилирован из файлов папки SRC (не складывайте хедеры в отдельный каталог).

Компилятор `g++ -std=c++14`.

Примечание для Python.

Python 3.7.3

Примечание для Java.

Java(TM) SE Runtime Environment (build 1.8.0_211-b12)

task1

Реализуйте функцию, которая конвертирует число (без знака) из десятичной системы исчисления в любую другую. Ваша функция должна иметь следующий прототип:

`String itoBase(unsigned int nb, String base);` nb – это подаваемое число, base – система исчисления.

На пример, «01» - двоичная, «012» - троичная, «0123456789abcdef» - шеснадцатиричная, «котики» - система исчисления в котиках.

Дополнительно*: перегрузите функцию, чтобы она могла конвертировать число из любой системы исчисления в любую другую:

`String itoBase(String nb, String baseSrc, String baseDst);`

Для проверки задания, напишите метод main, который принимает необходимые значения из аргументов командной строки, и выводит результат на экран. При некорректном вводе аргументов должен выводиться usage.

task2

Напишите программу, которая находит точки столкновения сферы и прямой линии. Если их нет, то выводится фраза: «Коллизий не найдено» (кириллицей, будьте внимательны), если есть, то выводятся координаты точек, ограниченные символом новой строки. Координаты считываются из файла, который имеет следующий формат:

```
{sphere: {center: [0, 0, 0], radius: 10.67}, line: {[1, 0.5, 15], [43, -14.6, 0.04]}}
```

Примечание: файл не будет содержать синтаксических ошибок, однако объекты и ключи могут находится в свободной последовательности. Координаты точек – массив [x, y, z].

Дополнительно: верх крутости – рендеринг данной сцены.

task3

Некоторое количество человек то наливают воду в бочку, то черпают из бочки. Если человек пытается налить больше воды, чем есть свободного объема – это ошибка, при этом объем воды в бочке не меняется. Так же если человек пытается зачерпнуть больше воды, чем есть в бочке – ошибка, объем воды также при этом не меняется. В остальных случаях – успех.

Вам дан лог файл. Напишите программу, которая ответит на следующие вопросы:

- какое количество попыток налить воду в бочку было за указанный период?
- какой процент ошибок был допущен за указанный период?
- какой объем воды был налит в бочку за указанный период?
- какой объем воды был не налит в бочку за указанный период?
- ... тоже самое для забора воды из бочки ...
- какой объем воды был в бочке в начале указанного периода? Какой в конце указанного периода?

Путь к логу, желаемый период – подаются в качестве аргументов командной строки. Результат записывается в csv файл (с наименованием столбцов).

Пример строки запуска: `java -jar App ./log.log 2020-01-01T12:00:00 2020-01-01T13:00:00`

Пример лог файла:

META DATA:

200 (объем бочки)

32 (текущий объем воды в бочке)

2020-01-01T12:51:32.124Z – [username1] - wanna top up 10l (успех)

2020-01-01T12:51:34.769Z – [username2] - wanna scoop 50l (фейл)

...

Примечание: для проверки сгенерируйте лог файл объемом 1 Mb, приложите его к решению.

Обратите внимание, искомого временного интервала может не быть в логе, приложение не должно при этом крашиться. Если аргументы поданы не верно, в stdout должен выводиться usage.

task4

Напишите программу, которая сравнивает 2 строки одинаковые ли они. Результат: вывод «OK», если строки идентичны, «KO», если не идентичны. Строки подаются в виде аргументов командной строки.

Примечание: во второй строке может быть символ '*' – он заменяет собой любое количество любых символов.

На пример:

«aaaa» «aaaa» - OK

«aaaa» «aa*» - OK

«a» «*****» - OK

Дополнение: конечно, в этом задании нас будет интересовать ваш код.