Тестовое задание для нового сотрудника

отдела Сборка.

1. Есть ресурс «https://yandex.com/time/sync.json?geo=213», что сделать:
   1. написать скрипт, который выполняет запрос к данному ресурсу, получает ответ, результат ответа выводит на экран в сыром виде.
   2. значение времени в «человекопонятном» формате и название временной зоны
   3. выводит дельту времени между точкой перед началом выполенения запроса и результатом из ответа ресурса о текущем времени с учетом часового пояса
   4. замеры из пункта c) повторить серией из пяти запросв и вывести среднюю дельту на основе данной серии
2. Написать универсальный «сборочный» скрипт, на входе адрес репозитория, относительный путь внутри репозитория до исходного кода, версия будущего продукта.
   1. Что делает скрипт:
      1. Выкачивает исходники по адресу репозитория
      2. Удаляет все директории в корне, кроме директории исходного кода(параметр путь до исходного кода)
      3. По пути расположения исходного кода создает служебный файл «version.json» следующего содержания: { "name": "hello world ", "version": "<тут версия из параметров>", "files": [<тут список файлов в дирекnории исходного кода с расширения \*.py, \*.js, \*.sh>] }
      4. Упакует в архив исходный код и служебный файл, имя архива получить как последнее имя диреткории в пути расположения исходного кода с добавлений текущей даты без разделителей
   2. На примере <https://github.com/paulbouwer/hello-kubernetes>
      1. Путь src/app
      2. Версия 25.3000
      3. { "name": "hello world ", "version": "25.3000", "files": [“server.js”] }
      4. Архив app01012024.zip
   3. Все действия сопровождать сообщения в консоль(логирование процесса) с указанием времени исполнения
3. Написать универсальный скрипт, на входе номер версии продукта и имя конфигурационного файла, в котором указаны шаблоны нумерации версий условных сервисов (пример ниже). Что делате скрипт: считывает конфигурационный файл, на основе шаблонов генерирует номера версий(на каждый шаблон по два варианта номера). Далее выводит отсортированный список всех полученных номеров, а потом список номеров меньше(старее) версии из параметров запуска скрипта.

Шаблон соответствует следующему общепринятому формату - числа разделенные точкой, символ \* означает элемент генерации. Пример конфигурационного файла:

{ Sh1:”3.7.\*”, Sh2:”3.\*.1”, Sh3:”1.2.3.\*”}, например Sh1 удовлетворяет 3.7.7 и 3.7.3

Количество ключей(шаблонов) в файле неограничено.

1. В системе зарегистрировано и запущено несколько systemd юнитов с именем "foobar-название\_сервиса". Каждый запускает определенный сервис. В каждом юните есть параметр с рабочей директорией вида:   
   WorkingDirectory=/opt/misc/название\_сервиса  
   и параметр ExecStart вида  
   ExecStart=/opt/misc/название\_сервиса/foobar-daemon произвольные\_параметры  
   Написать скрипт который получит список юнитов с таким именем поочередно их остановит, перенесет файлы сервиса из /opt/misc/название\_сервиса в /srv/data/название\_сервиса, поправит пути в параметрах WorkingDirectory и ExecStart и запустит эти юниты

Правила:

* Код заданий 1-3 реализовать на Python
* Код задания 4 реализовать на bash
* Код выложить на github.com или gitlab.com, прислать ссылку