Тестовые задания

Задания выполняются с использованием Python, версии не ниже 3.6 (желательно 3.8 - это наша рабочая версия для всех новых проектов). Выбор дополнительных библиотек - на усмотрение исполнителя.

Сканер портов

Для проверки безопасности сетей, а также для сбора информации о работающих сетевых сервисах применяется процесс, называемый сканированием сети. В рамках задачи необходимо разработать консольное приложение, принимающее на вход диапазон ip-адресов (например 192.168.1.0/24) и список портов (например 80, 443, 22, 21, 25). Результатом выполнения должен быть список открытых портов с указанием удаленного хоста. Пример вывода приведен в листинге 1.

```
192.168.1.1 80 OPEN
192.168.1.1 443 OPEN
192.168.1.4 22 OPEN
192.168.1.11 22 OPEN
192.168.1.12 25 OPEN
192.168.1.5 22 OPEN
192.168.1.8 25 OPEN
```

Листинг 1 - Пример отчета приложения «Сканер портов»

Так как диапазон для сканирования может быть весьма широкий, необходимо предусмотреть логику параллельного выполнения.

Дополнительное задание:

Как правило, порты 80 и 443 слушает веб-сервер. В некоторых случаях в заголовке ответа в поле Server передается название службы. Пример ответа, содержащего такую информацию приведен в листинге 2.

```
$ curl -X HEAD -I http://mail.ru

HTTP/1.1 301 Moved Permanently
Server: nginx/1.10.3
Date: Wed, 17 Jan 2018 07:56:43 GMT
Content-Type: text/html
Content-Length: 37
Connection: keep-alive
```

Листинг 2 - Пример ответа на http-запрос, который содержит поле server

В рамках дополнительного задания необходимо реализовать логику определения ПО сервера, слушающего порты 80 и 443. Разумеется, запрос должен посылаться только в том случае, если один из указанных портов на удаленном хосте открыт. Также необходимо учитывать, что данное поле присутствует в заголовке не всегда (это зависит от настроек сервера).

Подбор похожих доменов

Для оперативного поиска фишинговых ресурсов может применяться следующая логика:

- составляется первичный набор ключевых слов, ассоциирующихся с целевой компанией
- при помощи набора стратегий (например, одна из них — подстановка схожих по написанию символов) формируется расширенный набор ключевых слов
- полученное на предыдущем шаге множество перемножается на некоторое множество доменных зон (ru, com, net, org, biz и т. п.)
- отправляются dns-запросы с целью получить IP-адрес по каждому из элементов списка
- домены, по которым удалось определить ip, попадают в отчет

В рамках задания необходимо разработать консольное приложение, решающее описанную выше задачу. Входные данные — набор ключевых слов, результат — список доменов с ip-адресом. Стратегии формирования набора ключевых слов описаны в табл.1. Список доменных зон для подстановки представлен ниже. DNS-запросы должны отправляться параллельно.

Стратегия	**Входное слово**	**Выходные слова**
Добавление одного символа в конец строки	group-ib	group-iba group-ibb group-ibc ...
Подстановка символа, схожего по написанию (homoglyph)	group-ib	gr0up-ib group-1b gr0up-1b
Выделение поддомена, т. е. добавление точки	group-ib	group-i.b grou.p-ib gro.up-ib gr.oup-ib g.roup-ib
Удаление одного символа	group-ib	group-i group-b groupib

		grou-ib ...

Таблица 1 — Стратегии формирования ключевых слов

Список доменных зон:

com, ru, net, org, info, cn, es, top, au, pl, it, uk, tk, ml, ga, cf, us, xyz, top, site, win, bid

Парсер магазина приложений Google Play

В целях защиты бренда часто возникает задача найти мобильные приложения, содержащие определенное ключевое слово. Как правило, магазины не предоставляют API для этих целей, вследствие чего приходится парсить сайт.

Для Google Play можно получить список приложений по ключевому слову «сбербанк» на следующей странице: https://play.google.com/store/search?q=%D1%81%D0%B1%D0%B5%D1%80%D0%B1%D0%B0%D0%BD%D0%BA&c=apps

В рамках задачи необходимо разработать консольное приложение, которое осуществляет поиск по заданному ключевому слову и возвращает информацию о найденных приложениях в виде объекта json.

Для каждого найденного приложения должна быть возвращена следующая информация:

- название
- url страницы приложения
- автор
- категория
- описание
- средняя оценка
- количество оценок
- последнее обновление

При выполнении задания учитывать следующие моменты:

- Для популярных ключевых слов результат может быть разбит на несколько страниц. Необходимо обработать все страницы.
- У некоторых приложений искомое ключевое слово не присутствует ни в названии, ни в описании. Например у приложения «Аксиома» (https://play.google.com/store/apps/details?id=com.aksioma.aksiomapitanie) в указанных выше полях нет слова «сбербанк», но в поисковую выдачу приложение попадает. Такие приложения **не должны** попадать в отчет.

Дополнительное задание:

Как видно на примере поиска по слову «сбербанк», по некоторым ключевым словам результатов (а следовательно и страниц для последующей загрузки) может быть несколько сотен. В «боевых» условиях все запросы отправляются через медленные прокси, поэтому последовательная обработка страниц может занять много времени. Решается это параллельными запросами. В рамках доп задания нужно реализовать такую параллельную обработку.