

Часть А

В этой задаче и последующих её версиях вам нужно реализовать систему хранения транспортных маршрутов и обработки запросов к ней. Сначала на вход подаются запросы на создание базы данных, затем — запросы к самой базе.

При проектировании решения учитывайте, что его придётся неоднократно расширять. Авторские решения вы увидите лишь после успешной сдачи всех версий: таким образом, на протяжении всей череды доработок функционала вы будете иметь дело со своей собственной архитектурой кода и почувствуете, какие именно проектировочные решения мешают расширению.

Не пренебрегайте юнит-тестами! Именно они дадут вам уверенность в том, что при изменении части функционала не сломается всё остальное.

Как и в других задачах, здесь можно отправить на проверку архив из нескольких файлов.

Формат ввода базы данных

В первой строке вводится количество запросов на обновление базы данных, затем — по одному на строке — вводятся сами запросы. Запросы бывают двух типов.

Stop X: latitude, longitude

Остановка с названием **X** и координатами **latitude** и **longitude**.

Гарантируется, что остановка определена не более чем в одном запросе **Stop X**.

Bus X: описание маршрута

Запрос на добавление автобуса номер **X**. **Описание маршрута** может задаваться в одном из двух форматов (см. пример):

1. **stop1 - stop2 - ... - stopN**: автобус следует от **stop1** до **stopN** и обратно с указанными промежуточными остановками.
2. **stop1 > stop2 > ... > stopN > stop1**: кольцевой маршрут с конечной **stop1**.

По сути первая версия описания маршрута эквивалентна **stop1 > stop2 > ... > stopN-1 > stopN > stopN-1 > ... > stop2 > stop1**.

Гарантируется, что каждая из остановок маршрута определена в некотором запросе **Stop**, а сам маршрут **X** определён не более чем в одном запросе **Bus**.

Формат запросов к базе данных

В первой строке вводится количество запросов к базе данных, затем — по одному на строке — вводятся сами запросы. Запросы бывают (пока что) только одного типа.

Bus X

Вывести информацию об автобусе **X** в следующем формате:

Bus X: R stops on route, U unique stops, L route length

- **R** — количество остановок в маршруте автобуса от **stop1** до **stop1** включительно.
- **U** — количество уникальных остановок, на которых останавливается автобус. Одинаковыми считаются остановки, имеющие одинаковые названия.
- **L** — длина маршрута в метрах. Для простоты будем считать, что автобус проезжает путь между двумя соседними остановками по кратчайшему расстоянию по земной поверхности. При вычислении расстояния между двумя точками на земной поверхности считайте число π равным 3.1415926535, а радиус Земли — 6371 км.

Величину расстояния (а также другие вещественные числа в последующих частях задачи) выводите с 6 значащими цифрами, то есть предваряя модификатором `setprecision(6)`. (При этом ваши значения сравниваются с верными не посимвольно, а с относительной погрешностью 0,0001.)

Если автобус **X** не найден, выведите **Bus X: not found**

Ограничения

- 1 секунда на обработку всех запросов.
- Не более 2000 запросов на создание базы.
- Не более 2000 запросов к уже созданной базе.
- Не более 100 остановок в маршруте.
- Не более 25 символов в названии автобуса или остановки. Допустимые символы — латинские буквы, цифры и пробелы. Все названия непусты, не могут начинаться на пробелы или заканчиваться ими.

Пример

Ввод

10

Stop Tolstopaltsevo: 55.611087, 37.20829

Stop Marushkino: 55.595884, 37.209755

Bus 256: Biryulyovo Zapadnoye > Biryusinka > Universam > Biryulyovo Tovarnaya > Biryulyovo Passazhirsкая > Biryulyovo Zapadnoye

Bus 750: Tolstopaltsevo - Marushkino - Rasskazovka

Stop Rasskazovka: 55.632761, 37.333324

Stop Biryulyovo Zapadnoye: 55.574371, 37.6517

Stop Biryusinka: 55.581065, 37.64839

Stop Universam: 55.587655, 37.645687

Stop Biryulyovo Tovarnaya: 55.592028, 37.653656

Stop Biryulyovo Passazhirsкая: 55.580999, 37.659164

3

Bus 256

Bus 750

Bus 751

Вывод

Bus 256: 6 stops on route, 5 unique stops, 4371.02 route length

Bus 750: 5 stops on route, 3 unique stops, 20939.5 route length

Bus 751: not found

Часть В

Реализуйте предыдущую версию задачи со следующими изменениями.

Формат запросов к базе данных: изменения

Stop X(новый запрос)

Вывести информацию об остановке **X** в следующем формате:

Stop X: buses bus1 bus2 ... busN

- **bus1 bus2 ... busN** — список автобусов, проходящих через остановку. Дубли не допускаются, названия должны быть отсортированы в алфавитном порядке.
- Если остановка **X** не найдена, выведите **Stop X: not found**
- Если остановка **X** существует в базе, но через неё не проходят автобусы, выведите **Stop X: no buses**

Пример

Ввод

13

Stop Tolstopaltsevo: 55.611087, 37.20829

Stop Marushkino: 55.595884, 37.209755

Bus 256: Biryulyovo Zapadnoye > Biryusinka > Universam > Biryulyovo Tovarnaya
> Biryulyovo Passazhirsкая > Biryulyovo Zapadnoye

Bus 750: Tolstopaltsevo - Marushkino - Rasskazovka

Stop Rasskazovka: 55.632761, 37.333324

Stop Biryulyovo Zapadnoye: 55.574371, 37.6517

Stop Biryusinka: 55.581065, 37.64839

Stop Universam: 55.587655, 37.645687

Stop Biryulyovo Tovarnaya: 55.592028, 37.653656

Stop Biryulyovo Passazhirskaya: 55.580999, 37.659164

Bus 828: Biryulyovo Zapadnoye > Universam > Rossoshanskaya ulitsa > Biryulyovo Zapadnoye

Stop Rossoshanskaya ulitsa: 55.595579, 37.605757

Stop Prazhskaya: 55.611678, 37.603831

6

Bus 256

Bus 750

Bus 751

Stop Samara

Stop Prazhskaya

Stop Biryulyovo Zapadnoye

Вывод

Bus 256: 6 stops on route, 5 unique stops, 4371.02 route length

Bus 750: 5 stops on route, 3 unique stops, 20939.5 route length

Bus 751: not found

Stop Samara: not found

Stop Prazhskaya: no buses

Stop Biryulyovo Zapadnoye: buses 256 828

Часть С

Реализуйте вторую версию задачи со следующими изменениями.

В базе появляются данные о реальном — измеренном по дорогам — расстоянии между остановками.

Stop X (изменения запроса на добавление остановки)

Новый формат запроса на добавление остановки: **Stop X: latitude, longitude, D1m to stop1, D2m to stop2, ...**

А именно, теперь после широты и долготы может содержаться список расстояний от этой остановки до соседних. По умолчанию предполагается, что расстояние от **X** до **stop#** равно

расстоянию от **stop#** до **X**, если только расстояние от **stop#** до **X** не задано явным образом при добавлении остановки **stop#**

Гарантируется, что каждая из остановок **stop#** определена в некотором запросе **Stop**.

В рамках одного запроса **Stop** все **stop#** различны, и их количество не превосходит 100.

Все **Di** являются целыми положительными числами, каждое из которых не превышает 1000000 и не меньше расстояния по прямой между соответствующими остановками. После каждого расстояния обязательно следует буква **m**.

Bus X (изменения запроса на информацию об автобусе)

Новый формат ответа на запрос информации об автобусе: **Bus X: R stops on route, U unique stops, L route length, C curvature**

- **L** теперь вычисляется с использованием дорожного расстояния, а не географических координат.
- **C** (извилистость) — отношение длины маршрута, вычисленной с помощью дорожного расстояния, (новое **L**) к длине маршрута, вычисленной с помощью географического расстояния (**L** из предыдущих версий задачи). Таким образом, **C** — вещественное число, большее единицы. Оно может быть равно единице только в том случае, если автобус едет между остановками по кратчайшему пути (и тогда больше похож на вертолёт), а меньше единицы может оказаться только благодаря телепортации или хордовым тоннелям.

Гарантируется, что для любых двух соседних остановок любого маршрута так или иначе задано расстояние по дорогам.

Пример

Ввод

22

13

Stop Tolstopaltsevo: 55.611087, 37.20829, 3900m to Marushkino

Stop Marushkino: 55.595884, 37.209755, 9900m to Rasskazovka

Bus 256: Biryulyovo Zapadnoye > Biryusinka > Universam > Biryulyovo Tovarnaya > Biryulyovo Passazhirsкая > Biryulyovo Zapadnoye

Bus 750: Tolstopaltsevo - Marushkino - Rasskazovka

Stop Rasskazovka: 55.632761, 37.333324

Stop Biryulyovo Zapadnoye: 55.574371, 37.6517, 7500m to Rossoshanskaya ulitsa, 1800m to Biryusinka, 2400m to Universam

Stop Biryusinka: 55.581065, 37.64839, 750m to Universam

Stop Universam: 55.587655, 37.645687, 5600m to Rossoshanskaya ulitsa, 900m to Biryulyovo Tovarnaya

Stop Biryulyovo Tovarnaya: 55.592028, 37.653656, 1300m to Biryulyovo Passazhirsкая

Stop Biryulyovo Passazhirskaya: 55.580999, 37.659164, 1200m to Biryulyovo Zapadnoye

Bus 828: Biryulyovo Zapadnoye > Universam > Rossoshanskaya ulitsa > Biryulyovo Zapadnoye

Stop Rossoshanskaya ulitsa: 55.595579, 37.605757

Stop Prazhskaya: 55.611678, 37.603831

6

Bus 256

Bus 750

Bus 751

Stop Samara

Stop Prazhskaya

Stop Biryulyovo Zapadnoye

Вывод

Stop Samara: not found

Stop Prazhskaya: no buses

Stop Biryulyovo Zapadnoye: buses 256 828

Bus 256: 6 stops on route, 5 unique stops, 5950 route length, 1.361239 curvature

Bus 750: 5 stops on route, 3 unique stops, 27600 route length, 1.318084 curvature

Bus 751: not found

Часть D

В этой версии необходимо перевести весь ввод и вывод на формат JSON. Детали — в примере ниже.

Гарантируется, что вещественные числа не задаются в экспоненциальной записи, то есть обязательно имеют целую часть и, возможно, дробную часть через десятичную точку.

Каждый запрос к базе дополнительно получает идентификатор в поле **id** — целое число от 0 до 2147483647. Ответ на запрос должен содержать идентификатор этого запроса в поле **request_id**. Это позволяет выводить ответы на запросы в любом порядке.

Ключи словарей могут располагаться в произвольном порядке. Форматирование (то есть пробельные символы вокруг скобок, запятых и двоеточий) не имеет значения как во входном, так и в выходном JSON.

Часть E

В этой версии вам понадобится добавить в транспортный справочник функционал маршрутизации, реализованный с помощью предоставленных нами библиотек.

Изменения формата ввода

Новая секция — `routing_settings`

Во входной JSON добавляется ключ `"routing_settings"`, значением которого является словарь с двумя ключами:

- `"bus_wait_time"` — время ожидания автобуса на остановке (в минутах). Считайте, что когда бы человек ни пришёл на остановку и какой бы ни была эта остановка, он будет ждать любой автобус в точности указанное количество минут. Значение — целое число от 1 до 1000.
- `"bus_velocity"` — скорость автобуса (в км/ч). Считайте, что скорость любого автобуса постоянна и в точности равна указанному числу. Время стоянки на остановках не учитывается, время разгона и торможения — тоже. Значение — вещественное число от 1 до 1000.

Пример

```
"routing_settings": {  
  "bus_wait_time": 6,  
  "bus_velocity": 40  
}
```

Данная конфигурация задаёт время ожидания равным 6 минутам и скорость автобусов равной 40 километрам в час.

Новый тип запросов к базе — `Route`

В список `stat_requests` добавляются элементы с `"type": "Route"` — это запросы на построение маршрута между двумя остановками. Помимо стандартных свойств `"id"` и `"type"`, они содержат ещё два:

- `"from"` — остановка, в которой нужно начать маршрут.
- `"to"` — остановка, в которой нужно закончить маршрут.

Оба значения — названия существующих в базе остановок (однако, возможно, не принадлежащих ни одному автобусному маршруту).

Пример

```
{  
  "type": "Route",  
  "from": "Biryulyovo Zapadnoye",  
  "to": "Universam",  
  "id": 4  
}
```

Данный запрос означает построение маршрута от остановки «Biryulyovo Zapadnoye» до остановки «Universam».

На маршруте человек может использовать несколько автобусов, и даже один автобус несколько раз — если на некоторых участках он делает большой крюк и проще срезать на другом автобусе.

Маршрут должен быть наиболее оптимален по времени. Если маршрутов с минимально возможным суммарным временем несколько, допускается вывести любой из них: тестирующая система проверяет лишь совпадение времени маршрута с оптимальным и корректность самого маршрута.

При прохождении маршрута время расходуется на два типа активностей:

- Ожидание автобуса. Всегда длится `bus_wait_time` минут.
- Поездка на автобусе. Всегда длится ровно такое количество времени, которое требуется для преодоления данного расстояния со скоростью `bus_velocity`. Расстояние между остановками вычисляется по дорогам, то есть с помощью `road_distances`.

Ходить пешком, выпрыгивать из автобуса между остановками и использовать другие виды транспорта запрещается. На конечных остановках все автобусы высаживают пассажиров и уезжают в парк. Даже если человек едет на кольцевом (`"is_roundtrip": true`) маршруте и хочет проехать мимо конечной, он будет вынужден выйти и подождать тот же самый автобус ровно `bus_wait_time` минут. Этот и другие случаи разобраны в примерах.

Ответ на запрос `Route` устроен следующим образом:

```
{  
  "request_id": <id запроса>,  
  "total_time": <суммарное время>,  
  "items": [  
    <элементы маршрута>  
  ]  
}
```


`total_time` — суммарное время в минутах, требуемое для прохождения маршрута, выведенное в виде вещественного числа.

Обратите внимание, что расстояние от остановки А до остановки В может быть не равно расстоянию от В до А!

`items` — список элементов маршрута, каждый из которых описывает непрерывную активность пассажира, требующую временных затрат. А именно, элементы маршрута бывают двух типов.

Wait — подождать нужное количество минут (в нашем случае — всегда `bus_wait_time`) на указанной остановке:

```
{
  "type": "Wait",
  "stop_name": "Biryulyovo",
  "time": 6
}
```

Bus — проехать `span_count` остановок (перегонов между остановками) на автобусе `bus`, потратив указанное количество минут:

```
{
  "type": "Bus",
  "bus": "297",
  "span_count": 2,
  "time": 5.235
}
```

В случае отсутствия маршрута между указанными остановками выведите результат в следующем формате:

```
{
  "request_id": <id запроса>,
  "error_message": "not found"
}
```

