

اعداد یکتا

- محدودیت زمان: ۵/۰ ثانیه
- محدودیت حافظه: ۶۴ مگابایت

دنباله‌ای به طول n به فرم a_1, a_2, \dots, a_n از اعداد صحیح داریم. می‌دانیم هیچ عددی در این دنباله، بیش از ۲ بار ظاهر نشده است. برای مثال این دنباله می‌تواند به صورت 1, 3, 2, 8, 7, 2, 3, 4 باشد.

به یک عدد صحیح «یکتا» می‌گوییم اگر فقط یکبار در این دنباله ظاهر شده باشد. برای مثال، در دنباله بالا تنها اعداد 1, 4, 7 و 8 «یکتا» هستند.

از شما می‌خواهیم برنامه‌ای بنویسید که با دریافت اعداد موجود در این دنباله، XOR اعداد «یکتا» آن را چاپ کند. اگر هیچ عدد «یکتا»یی در این دنباله نبود، صفر چاپ کنید.

▼ اگر نمی‌دانید که XOR چیست، روی این قسمت کلیک کنید.

منظور از XOR (مخفف *exclusive OR*) دو عدد صحیح و نامنفی مثل a و b که آن را به صورت $a \oplus b$ نیز نشان می‌دهند، این است که اگر دو عدد a و b را در مبنای دو زیرهم بنویسیم. (اگر یکی از این اعداد تعداد کمتری رقم دارد پشت آن صفر در نظر بگیرید.) سپس برای هر دو رقم زیرهم اگر یکسان باشند رقم متناظر حاصل، صفر و در غیر این صورت یک خواهد بود.

برای مثال برای محاسبه $6 \oplus 12$ ابتدا این دو عدد را در مبنای دو می‌نویسیم (باید پشت ۶ صفر اضافه کنیم تا تعداد ارقام برابر شود). سپس به صورت رقم به رقم نگاه می‌کنیم و اگر ارقام متناظر در این دو عدد برابر بودند، ۰ و در غیر این صورت ۱ می‌گذاریم یعنی:

$$6 \oplus 12 = (0110)_2 \oplus (1100)_2 = (1010)_2 = 10$$

همچنین از تعریف مشخص است که این عمل، خاصیت «جابه‌جایی» و «شرکت‌پذیری» دارد. یعنی اگر m عدد داشته باشیم، ترتیب این اعداد و یا ترتیب عملیات‌ها بر روی حاصل نهایی تاثیری ندارد. بنابراین پاسخ مسئله فقط یک حالت دارد.

اگر می‌خواهید درباره این عملگر بیشتر بدانید، این پیوند را مطالعه کنید.

ورودی

در خط اول ورودی، عدد طبیعی n نوشته می‌شود.

$$1 \leq n \leq 100$$

در خط دوم، اعداد آرایه که با یک فاصله از یکدیگر جدا شده‌اند نوشته می‌شوند.

$$1 \leq a_i \leq 100$$

خروجی

در تنها سطر خروجی، XOR اعداد «یکتا»ی این دنباله را چاپ کند. اگر هیچ عدد «یکتا»یی در این دنباله نبود، 0 چاپ کنید.

ورودی نمونه ۱

2
1 1

خروجی نمونه ۱

0

هیچ عدد یکتایی در دنباله 1, 1 وجود ندارد، بنابراین پاسخ این نمونه ۰ خواهد بود.

ورودی نمونه ۲

8

4 3 2 7 8 2 3 1

خروجی نمونه ۲

10

اعداد یکتای دنباله 1, 2, 3, 4, 7, 8, 2, 3, 4, 3, 2, 4, 1, 7 و 8 هستند بنابراین پاسخ این نمونه برابر است با:

$$1 \oplus 4 \oplus 7 \oplus 8 = 10$$

ورودی نمونه ۳

2

2 1

خروجی نمونه ۳

3

همه اعداد دنباله 1, 2, 1, 2 هستند. بنابراین پاسخ این نمونه برابر است با:

$$2 \oplus 1 = 3$$

دیجی رمز

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

سال ۱۴۹۹ است و دیجی کالا از رمزارز خود با نام دیجی رمز رونمایی کرده است!

علی می خواهد از طریق سرمایه گذاری در این رمزارز، m تومان پول کسب کند، اما در حال حاضر هیچ سرمایه ای ندارد. او می خواهد از دوستش پول قرض کند تا این سرمایه گذاری ها را انجام دهد.

در بازار دیجی رمز، n فرصت سرمایه گذاری وجود دارد که i امین آن ها ابتدا به c_i تومان پول برای شروع نیاز دارد و پس از سرمایه گذاری، هر روز p_i تومان سود می دهد. علی در هر یک از این فرصت ها می تواند حداکثر یک بار سرمایه گذاری کند، اما او می تواند در هر چند فرصت مختلفی که بخواهد سرمایه گذاری کند.

به علی کمک کنید تا روشی برای سرمایه گذاری انتخاب کند که در کوتاه ترین زمان بتواند تمام پول قرض گرفته شده از دوستش را به او پس بدهد و برای خودش هم حداقل m تومان پول بماند تا به هدفش برسد.

به او بگویید کوتاه ترین زمان چند روز است.

ورودی

در سطر اول ورودی دو عدد صحیح و مثبت n و m با فاصله از هم آمده است.

$$1 \leq n \leq 100\,000$$

$$1 \leq m \leq 10^9$$

در n سطر بعدی، در هر خط دو عدد p_i و c_i که با یک فاصله از هم جدا شده اند آمده است.

$$1 \leq p_i, c_i \leq 10^9$$

خروجی

در تنها سطر خروجی کمترین تعداد روزی که لازم است تا علی بعد از پس دادن پول دوستش m تومان سود کند را چاپ کنید.

مثال‌ها

ورودی نمونه ۱

2 5
4 10
10 15

خروجی نمونه ۱

2

اگر علی ۲ روز در فرصت سرمایه‌گذاری دوم سرمایه‌گذاری کند، به اندازه‌ی

$$2 \times 10 - 15 = 5 \geq 5$$

سود می‌کند و به هدفش می‌رسد.

ورودی نمونه ۲

4 10
1 8
3 12

4 17
10 100

خروجی نمونه ۲

6

اگر علی ۶ روز در فرصت سرمایه‌گذاری دوم و سوم سرمایه‌گذاری کند، به اندازه‌ی

$$(6 \times 3 - 12) + (6 \times 4 - 17) = 6 + 7 = 13 \geq 10$$

سود می‌کند و به هدفش می‌رسد.

ورودی نمونه ۳

3 5
4 1
9 10
6 3

خروجی نمونه ۳

1

اگر علی ۶ روز در فرصت سرمایه‌گذاری اول و سوم سرمایه‌گذاری کند، به اندازه‌ی

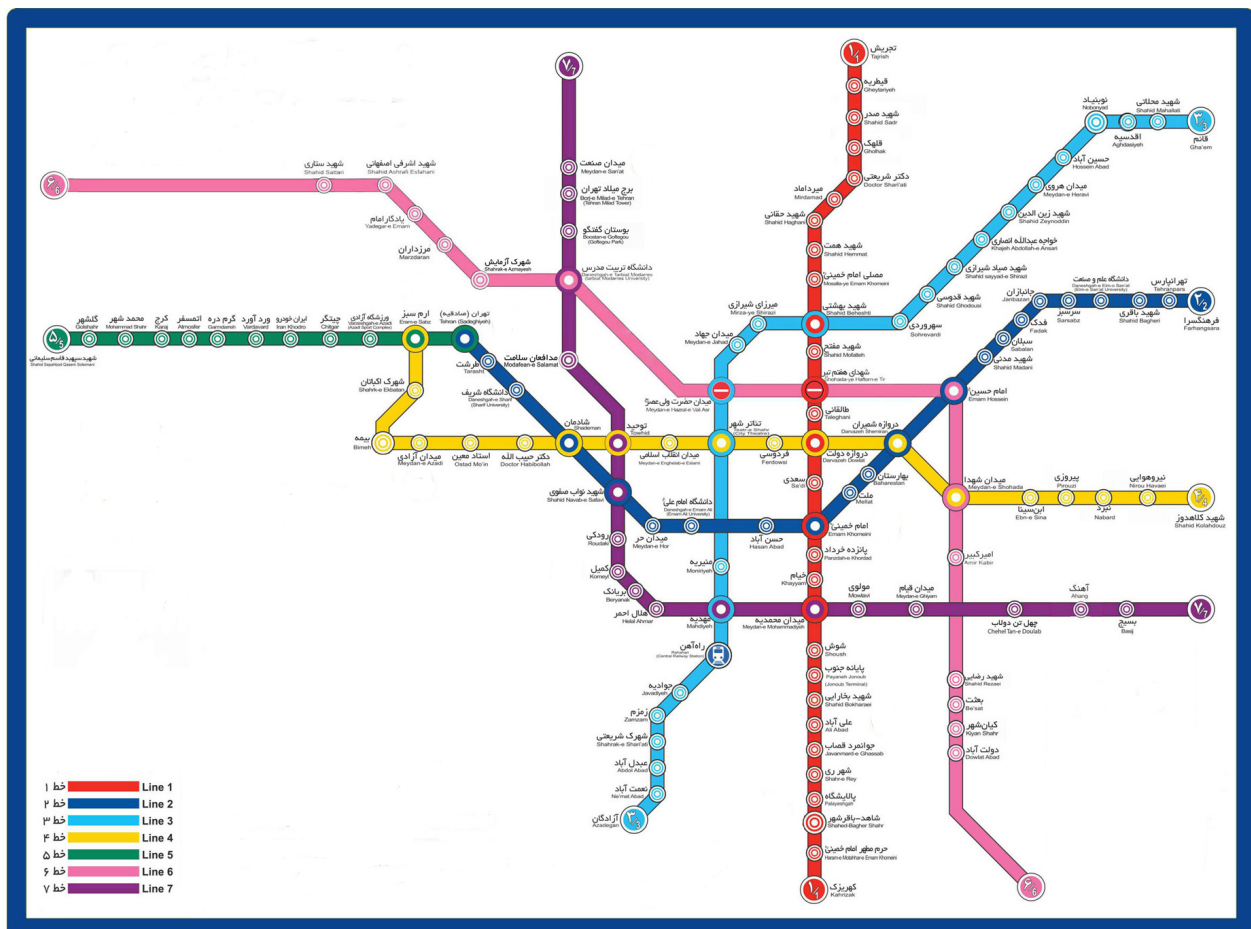
$$(1 \times 4 - 1) + (1 \times 6 - 3) = 3 + 3 = 6 \geq 5$$

سود می‌کند و به هدفش می‌رسد.

مترو

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

نقشه متروی تهران را در نظر بگیرید. (این عکس فقط برای دادن یک دید کلی به شماست و نیازی به خواندن کلمات آن نیست.)



اسم تمام ایستگاه‌ها در این فایل در ۷ سطر آمده است. در سطر i ام، نام ایستگاه‌های خط i ام مترو به ترتیب و با فاصله آمده است. (توجه کنید نام یک ایستگاه شامل کاراکتر فاصله ($space$) نیست.)

- زمان لازم تردد قطار بین دو ایستگاه ۲ دقیقه

- زمان لازم برای ورود به ایستگاهی که قصد سوار شدن قطار را داریم، ۱۷ دقیقه
- زمان لازم برای خارج شدن از ایستگاه که قصد پیاده زدن در آن را داریم، ۱۲ دقیقه.
- زمان لازم برای تعویض خط نیاز داریم، نیز ۱۰ دقیقه است.

توجه کنید در ایستگاه‌هایی که محل تقاطع دو خط هستند، می‌توان از طریق هر دو خط سوار شد و نیازی به معطلی برای تعویض خط نداریم.

محسن می‌خواهد خریدهای خودش را از دیجیکالا انجام نهد و با مترو برود و خرید کند. او بعد از مسیریابی فهمید که باید از ایستگاه s سوار مترو شود و در ایستگاه f پیاده شود.

از شما می‌خواهیم برنامه‌ای بنویسید که کمترین زمانی که طول می‌کشد تا محسن به محل خرید برسد را محاسبه کنید.

ورودی

در سطر اول ورودی عدد صحیح و مثبت q آمده است.

$$1 \leq q \leq 16000$$

در q سطر بعدی در هر سطر دو رشته s و f آمده که نام ایستگاه شروع و پایان را نشان می‌دهد.

تضمین می‌شود این دو رشته در فایل ایستگاه‌ها موجوداند.

خروجی

برای هر کدام از q جفت ایستگاه ورودی داده شده، کمینه زمان لازم برای رسیدن از ایستگاه مبدا به ایستگاه مقصد را محاسبه کنید.

مثال

ورودی نمونه ۱

5

Tajrish Gheydariyeh

Tajrish Gholhak

Tarasht Eram-e_Sabz

Aghdasiyeh Sohrevardi

Towhid Baharestan

خروجی نمونه ۱

31

35

43

45

51

سفر اول از «تجریش» به «قیطریه»:

- ۱۷ دقیقه ورود و سوار قطار شدن در ایستگاه «تجریش» طول می کشد.
- ۲ دقیقه طول می کشد تا در خط ۱، از ایستگاه «تجریش» به «قیطریه» برسد.
- ۱۲ دقیقه طول می کشد تا از ایستگاه «قیطریه» خارج شود.

بنابراین مجموع زمان این سفر برابر است با:

$$17 + 2 + 12 = 31$$

سفر دوم از «تجریش» به «قلهک»:

- ۱۷ دقیقه ورود و سوار قطار شدن در ایستگاه «تجریش» طول می کشد.
- ۶ دقیقه طول می کشد تا در خط ۱، از ایستگاه «تجریش» به «قلهک» برسد.
- ۱۲ دقیقه طول می کشد تا از ایستگاه «قلهک» خارج شود.

بنابراین مجموع زمان این سفر برابر است با:

$$17 + 6 + 12 = 35$$

سفر سوم از «طرشت» به «ارم سبز»:

- ۱۷ دقیقه ورود و سوار قطار شدن در ایستگاه «طرشت» طول می کشد.
- ۲ دقیقه طول می کشد تا در خط ۲، از ایستگاه «طرشت» به «تهران (صادقیه)» برسد.
- ۱۰ دقیقه طول می کشد تا از خط ۲ به ۵ در ایستگاه «تهران (صادقه)» تغییر مسیر بدهد.
- ۲ دقیقه طول می کشد تا در خط ۵، از ایستگاه «تهران (صادقیه)» به «ارم سبز» برسد.
- ۱۲ دقیقه طول می کشد تا از ایستگاه «ارم سبز» خارج شود.

بنابراین مجموع زمان این سفر برابر است با:

$$17 + 2 + 10 + 2 + 12 = 43$$

سفر چهارم از «اقدیسه» به «سهروردی»:

- ۱۷ دقیقه ورود و سوار قطار شدن در ایستگاه «اقدسیه» طول می کشد.
- ۱۶ دقیقه طول می کشد تا در خط ۳، از ایستگاه «اقدسیه» به «سهروردی» برسد.
- ۱۲ دقیقه طول می کشد تا از ایستگاه «سهروردی» خارج شود.

بنابراین مجموع زمان این سفر برابر است با:

$$17 + 16 + 12 = 45$$

سفر پنجم از «توحید» به «بهارستان»:

- ۱۷ دقیقه ورود و سوار قطار شدن در ایستگاه «توحید» طول می کشد.
- ۱۰ دقیقه طول می کشد تا در خط ۴، از ایستگاه «توحید» به «دروازه شمیران» برسد.
- ۱۰ دقیقه طول می کشد تا از خط ۴ به ۲ در ایستگاه «دروازه شمیران» تغییر مسیر بدهد.
- ۲ دقیقه طول می کشد تا در خط ۵، از ایستگاه «دروازه شمیران» به «بهارستان» برسد.

• ۱۲ دقیقه طول می‌کشد تا از ایستگاه «بهارستان» خارج شود.

بنابراین مجموع زمان این سفر برابر است با:

$$17 + 10 + 10 + 2 + 12 = 51$$

انبارگردانی

کوئری‌های شما باید روی آخرین نسخه‌ی *MySQL* قابل اجرا باشند.

محمد به تازگی به تیم مهندسی دیجی کالا پیوسته. او در یکی از تسک‌هایش لازم دارد تا کوئری‌های مختلفی برای کار با داده‌هایی که با آن‌ها سروکار دارد بنویسد. داده‌هایی که محمد با آن‌ها سروکار دارد مربوط به محصولات و سفارش‌های مربوط به آن‌ها است. او یک *denormalization* روی داده‌ها اعمال کرده، به این صورت که علاوه بر اطلاعات سفارش‌های محصولات، مجموع فروش هر محصول نیز در جدول اطلاعات محصولات وجود دارد.

از آنجایی که محمد فرصت نوشتن کوئری‌های مدنظرش را ندارد، از شما می‌خواهیم تا این کوئری‌ها را برایش بنویسید.

جزئیات پروژه

داده‌های اولیه برای تست نمونه را از این لینک دانلود کنید.

ساختار جداول به شرح زیر است:

جدول products : از این جدول برای نگهداری اطلاعات محصولات استفاده می‌شود. ساختار این جدول به صورت زیر است:

تعریف	نوع	نام ستون
شناسه‌ی محصول	BIGINT(20)	id
نام محصول	VARCHAR(255)	name
توضیحات محصول	TEXT	description
قیمت محصول	DECIMAL(15, 2)	price

تعریف	نوع	نام ستون
مجموع مبلغ فروش محصول	DECIMAL(15, 2)	total_profit
زمان درج محصول	TIMESTAMP	created_at

جدول orders : از این جدول برای نگهداری لیست سفارش‌ها استفاده می‌شود. ساختار این جدول به صورت زیر است:

تعریف	نوع	نام ستون
شناسه‌ی سفارش	BIGINT(20)	id
شناسه‌ی کاربر سفارش‌دهنده	BIGINT(20)	user_id
زمان ایجاد سفارش	TIMESTAMP	created_at

جدول order_details : از این جدول برای نگهداری اطلاعات اقلام موجود در سفارش‌ها استفاده می‌شود. ساختار این جدول به صورت زیر است:

تعریف	نوع	نام ستون
شناسه‌ی اطلاعات محصول در سفارش	BIGINT(20)	id
شناسه‌ی سفارش	BIGINT(20)	order_id
شناسه‌ی محصول	BIGINT(20)	product_id
تعداد سفارش داده شده‌ی محصول	INT(11)	quantity

جدول delivery_centers : از این جدول برای نگهداری اطلاعات مراکز ارسال استفاده می‌شود. ساختار این جدول به صورت زیر است:

تعریف	نوع	نام ستون
-------	-----	----------

تعریف	نوع	نام ستون
شناسه‌ی مرکز ارسال	BIGINT(20)	id
نام مرکز ارسال	VARCHAR(255)	name

جدول deliveries : از این جدول برای نگهداری اطلاعات تحویل سفارش‌ها استفاده می‌شود. ساختار این جدول به صورت زیر است:

نام ستون	نوع	تعریف
order_id	BIGINT(20)	شناسه‌ی سفارش
delivery_center_id	BIGINT(20)	شناسه‌ی مرکز ارسال
received_at	TIMESTAMP	زمان دریافت محصولات سفارش از انبار
delivered_at	TIMESTAMP	زمان تحویل سفارش به مشتری (در صورتی که سفارش هنوز تحویل داده نشده باشد، مقدار آن برابر با NULL خواهد بود.)

توجه داشته باشید که جدول کاربران صرفاً جهت سادگی در سؤال نیامده است. نیازی به اطلاعات چنین جدولی نیست.

مطلوبات

کوئری‌های زیر را طوری بنویسید که خروجی خواسته شده به دست آید:

۱. محمد اکنون متوجه شده که یک باگ در بک‌اند برنامه‌اش وجود داشته و مقدار ستون `total_profit` در جدول `products` لزوماً مطابق با اطلاعات موجود در جداول `orders` و `order_details` نیست (در داده‌ها *inconsistency* وجود دارد). کوئری‌ای بنویسید که مقدار ستون `total_profit` جدول `products` را طبق مقادیر موجود در جداول `orders` و `order_details` به روزرسانی کند تا *inconsistency* در صورت وجود رفع شود.

۲. کوئری‌ای بنویسید که شناسه‌ی ۵ مرکز ارسال برتر از نظر چابک بودن را برگرداند. یک مرکز ارسال در صورتی چابک است که میانگین اختلاف زمانی بین دریافت محصول از انبار تا تحویل آن به مشتری در سفارش‌های تحویل‌داده‌شده‌اش کمتر از سایر مراکز ارسال باشد. نتایج را به‌ترتیب نزولی میزان چابک بودن مرتب کنید. تضمین می‌شود که میزان چابک بودن مراکز ارسال با یکدیگر متفاوت است.

۳. کوئری‌ای بنویسید که شناسه‌ی محصولاتی که تا ۷ روز پس از انتشارشان کمتر از ۱۰ فروش داشتند را دریافت کند. نام ستون خروجی اهمیتی ندارد.

نکات

- کوئری‌های شما باید روی آخرین نسخه‌ی *MySQL* قابل اجرا باشند.
- کوئری هر بخش باید تنها شامل یک *statement* باشد.
- هر کوئری امتیاز جداگانه دارد و اگر کوئری یک قسمت را نتوانستید بنویسید، کوئری‌هایی که حل کردید را بفرستید و قسمت آن کوئری را خالی بگذارید.

آنچه باید آپلود کنید

پس از طراحی کوئری‌ها، آن‌ها را در قالب زیر در یک فایل با پسوند *.sql* آپلود کنید.

```

1 | -- Section1
2 |     Your first query here
3 | -- Section2
4 |     Your second query here
5 | -- Section3
6 |     Your third query here

```

بهینه سازی جست و جو

کوئری های شما باید روی آخرین نسخه ی *MySQL* قابل اجرا باشند.

در تیم مهندسی دیجی کالا، یک سیستم رده بندی برای سطح مهندسين وجود دارد. اميرحسين كه قصد ارتقاي درجه ي چند نفر از اعضاي تيم را دارد، تصميم گرفته تا سؤالي در خصوص ايندكس هاي ديتابيس از آن ها بپرسد و مطمئن شود كه آن ها درك خوبي از ديتابيس دارند. او سؤالات را از قبل آماده کرده، اما فرصت نوشتن پاسخ نمونه برای اين سؤالات را ندارد. از شما می خواهيم تا کوئري های مدنظر اميرحسين را بنويسيد.

جزئیات پروژه

داده های اولیه برای تست نمونه را از [این لینک](#) دانلود کنید.

ساختار جداول به شرح زیر است:

جدول users : از این جدول برای نگهداری اطلاعات کاربران استفاده می شود. ساختار این جدول به صورت زیر است:

تعريف	نوع	نام ستون
شناسه ی کاربر	BIGINT(20)	id
نام کاربری	VARCHAR(255)	username
نام کاربر	VARCHAR(255)	name
رمز عبور	VARCHAR(255)	password
زمان عضویت کاربر	TIMESTAMP	created_at

جدول **products** : از این جدول برای نگهداری اطلاعات محصولات استفاده می‌شود. فرض می‌شود که هر محصول تنها متعلق به یک دسته‌بندی است! ساختار این جدول به صورت زیر است:

تعریف	نوع	نام ستون
شناسه‌ی محصول	BIGINT(20)	id
شناسه‌ی دسته‌بندی محصول	BIGINT(20)	category_id
نام محصول	VARCHAR(255)	name
توضیحات محصول	TEXT	description
قیمت محصول	DECIMAL(15, 2)	price
زمان درج محصول	TIMESTAMP	created_at

توجه داشته باشید که جدول دسته‌بندی‌های محصولات صرفاً جهت سادگی در سؤال نیامده است. نیازی به اطلاعات چنین جدولی نیست.

مطلوبات

کوئری‌های خواسته‌شده از شما، موارد زیر است:

۱. کوئری ساخت ایندکس جهت بهینه‌سازی حداکثری سرعت دریافت شناسه‌ی کاربرانی که نام‌شان با یک رشته‌ی خاص آغاز می‌شود:

```
1 | SELECT id
2 | FROM users
3 | WHERE name LIKE 'Ali%'
```

۲. کوئری ساخت ایندکس جهت بهینه‌سازی حداکثری سرعت دریافت شناسه، نام و قیمت محصولات که قیمت‌شان بین دو عدد مشخص است و مربوط به یک دسته‌ی خاص هستند:

```
1 | SELECT id, name, price
2 | FROM products
3 | WHERE price > 25000 AND price < 50000000 AND category_id = 5
```

نکات

- کوئری‌های شما باید روی آخرین نسخه‌ی *MySQL* قابل اجرا باشند.
- کوئری هر بخش باید تنها شامل یک *statement* باشد.
- هر کوئری امتیاز جداگانه دارد و اگر کوئری یک قسمت را نتوانستید بنویسید، کوئری‌هایی که حل کردید را بفرستید و قسمت آن کوئری را خالی بگذارید.

آنچه باید آپلود کنید

پس از طراحی کوئری‌ها، آن‌ها را در قالب زیر در یک فایل با پسوند *.sql* آپلود کنید.

```
1 | -- Section1
2 |     Your first query here
3 | -- Section2
4 |     Your second query here
```

کوتاه کننده لینک

این سؤال تنها با زبان های *Go* ، *Python* ، *PHP* و *Node.js* (*JS*) قابل حل است.

دیجی کالا قصد دارد برای بخش تحویل محصولات خود یک سامانه ی ساده ی ثبت پیشنهادات و انتقادات راه اندازی کند. از شما می خواهیم یک *API* برای این سامانه طراحی کنید.

جزئیات پروژه

پروژه ی اولیه را از این لینک دانلود کنید.

در این سؤال، یک *REST API* شامل *endpoint* های زیر باید پیاده سازی شود:

عنوان	آدرس
بودن سرویس <i>up</i> بررسی	GET /
ثبت نام	POST /signup
ورود به حساب کاربری	POST /login
کوتاه کردن لینک	POST /urls
دریافت لیست لینک های کوتاه شده	GET /urls
لینک کوتاه شده	GET /{slug}

در این *API* هر کاربر باید یک توکن داشته باشد. این توکن برای هر کاربر ثابت است.

endpoint های مورد نیاز

بررسی *up* بودن سرویس

پاسخ این *endpoint* باید به صورت زیر باشد:

- کد وضعیت: 200

- بدنه: {"ok":true}

ثبت نام

دو پارامتر username و password باید به این endpoint ارسال شوند. در صورتی که حداقل یکی از این پارامترها ارسال نشده باشد یا برابر با رشته‌ی خالی باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400

- بدنه: {"ok":false,"error":"no username or password provied"}

اگر کاربری با نام کاربری وارد شده از قبل موجود باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400

- بدنه: {"ok":false,"error":"user already exists"}

در غیر این صورت، کاربر باید ساخته شود، یک توکن یکتا برایش تولید شود و پاسخ به صورت زیر باشد:

- کد وضعیت: 201

- بدنه: {"ok":true,"token":"USER_TOKEN"}

ورود به حساب کاربری

دو پارامتر username و password باید به این endpoint ارسال شوند. در صورتی که حداقل یکی از این پارامترها ارسال نشده باشد یا برابر با رشته‌ی خالی باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400

- بدنه: {"ok":false,"error":"no username or password provied"}

اگر نام کاربری یا رمز عبور نادرست باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400

- بدنه: {"ok":false,"error":"invalid username or password"}

در غیر این صورت، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 200
- بدنه: `{"ok":true,"token":"USER_TOKEN"}`

کوتاه کردن لینک

این *endpoint* نیازمند *authentication* است. مقدار *Authorization* باید برابر با توکن کاربر باشد (بدون **Bearer** یا موارد مشابه).

پارامتر *url* (لینک) باید به این *endpoint* ارسال شود. در صورتی که این پارامتر ارسال نشده باشد یا برابر با رشته‌ی خالی باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400
- بدنه: `{"ok":false,"error":"no url provied"}`

در غیر این صورت، لینک باید کوتاه شود و پاسخ به صورت زیر باشد (مقدار `{slug}` می‌تواند رندوم باشد):

- کد وضعیت: 201
- بدنه: `{"ok":true,"url":"http://localhost/{slug}"}`

دریافت لیست لینک‌های کوتاه‌شده

این *endpoint* نیازمند *authentication* است. مقدار *Authorization* باید برابر با توکن کاربر باشد (بدون **Bearer** یا موارد مشابه).

این *endpoint* باید لیست لینک‌های کوتاه‌شده‌ی کاربر به همراه تعداد بازدید هر کدام را در قالب یک لیست برگرداند.

- کد وضعیت: 200
- مثالی از پاسخ:

```
1  [
2    {
3      "short_url": "http://localhost/SDas2",
4      "url": "https://quera.org",
5      "visits_count": 3
6    },
7    {
8      "short_url": "http://localhost/zxcA54",
9      "url": "https://google.com",
10     "visits_count": 0
11   }
12 ]
```

لینک کوتاه‌شده

آدرس این درخواست به صورت `{slug} /` است که `{slug}` همان رشته‌ای است که به لینک کوتاه‌شده توسط برنامه تخصیص می‌یابد. اگر `{slug}` ورودی یافت نشود، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 404

در غیر این صورت، کاربر باید به لینک اصلی با کد پاسخ 301 هدایت شود.

نکات تکمیلی

- نیازی به *persistent* بودن داده‌ها نیست!

▼ نصب نیازمندی‌ها و اجرا

برای حل این سؤال می‌توانید از هر زبان و هر تکنولوژی‌ای که می‌خواهید استفاده کنید. به صورتی که در یک پوشه به نام `api` کد برنامه را نوشته و در فایل به نام `runner.sh` که توسط `sh` اجرا می‌شود، باید برنامه‌ی خود را اجرا کنید. توجه کنید که حتماً باید `Dockerfile` مربوط به پروژه‌ی خود را برای ما ارسال کنید.

در پروژه‌ی اولیه، ۴ داکر فایل برای php ، python ، golang و node قرار دادیم که می‌توانید از آن‌ها مستقیماً استفاده کنید. در صورتی که از یکی از این زبان‌ها برای حل سؤال استفاده می‌کنید، کافیهست که Dockerfile مربوط به آن را در پوشه‌ی api کپی کنید و طبق توضیحات داده شده، سؤال را حل کنید. برای نصب نیازمندی‌های پایتون از requirements.txt ، برای پی‌اچ‌پی از composer.json ، برای گولنگ از go.mod و برای نودجی‌اس از package.json استفاده کنید.

در صورتی که زبان مورد استفاده‌ی شما، چیزی به جز این ۴ مورد است، باید خودتان داکر فایلی در پوشه‌ی api به‌شکلی بنویسید که بتواند نیازمندی‌های پروژه‌ی شما را نصب کرده و برنامه‌ی شما را مانند داکر فایل‌های موجود اجرا کند.

▼ تغییر Dockerfile

امکان تغییر فایل Dockerfile وجود ندارد، اما در اسکریپت runner.sh می‌توانید هر دستوری را اجرا کنید.

نحوه ارسال پاسخ

شما می‌توانید تمامی محتوای موجود در پوشه‌ی api را تغییر دهید و هر فایلی که می‌خواهید اضافه یا کم کنید.

```

1 | api
2 | └─ api.py # or main.go somefile.js anyfile.php name.any ...
3 | └─ Dockerfile
4 | └─ requirements.txt # or go.mod package.json composer.json
5 | └─ runner.sh

```

توجه کنید که نام فایل کد شما برای سیستم داوری اهمیتی ندارد و این خود شما هستید که در runner.sh از نام آن برای اجرای پروژه استفاده می‌کنید.

در نهایت این پوشه را zip کرده و ارسال کنید. توجه کنید که پس از extract کردن فایل zip شما، باید پوشه‌ی api را ببینیم که درون آن Dockerfile وجود دارد.

