Lazy Wo/Man on Campus

- محدودیت زمان: 3 ثانیه
- محدودیت حافظه: 256 مگابایت

دانشجویی برای بهینه کردن رفت و آمد خود فرمولی ابداع کرده است تا همیشه گزینه‌ای که نیروی کمتری از او می‌برد را انتخاب کند؛ این فرمول *نیروی دانشجو* نام دارد. اگر فرض کنیم نیروی مصرفی دانشجو برای انتخابی F است، مقدار آن از ضرب مسافت ($distance$) در ضریب سختی ($coefficient$ $exhausting$) به دست می‌آید. یعنی:

$$F = d \times e$$

دانشجو در نقطه‌ی شروع (x_1, y_1) است و همیشه بین دو انتخاب رفتن به کلاس درس در مختصات (x_2, y_2) و یا رفتن به خوابگاه در مختصات (x_3, y_3) باید یکی را انتخاب کند. نیاز است شما برنامه‌ای بنویسید که برای این تصمیم گیری به او کمک کنید.

در نظر بگیرید ضریب سختی e هر کلاسی متفاوت است اما ضریب سختی رفتن به خوابگاه همیشه 1 است. همچنین دانشجو برای محاسبه‌ی فاصله‌ی d بین دو نقطه از فرمول اقلیدسی زیر استفاده می‌کند.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

بنابراین، برنامه‌ای بنویسید که با گرفتن سه مختصات شروع، کلاس و خوابگاه و ضریب سختی e به دانشجو بگوید که آیا باید به کلاس درس برود یا به خوابگاه! به عبارتی برنامه باید هر حالتی که *نیروی مصرفی دانشجو* را کمترین کند را چاپ کند.

نکته: دانشجو در صورتی که به برابری برسد برای غیبت نخوردن همیشه کلاس را انتخاب می‌کند.

بخش‌های برنامه

برنامه‌ی شما باید شامل دو تابع بوده و از ماژول $math$ استفاده کند.

ورودی

برنامه بایستی ورودی‌های x_1 و y_1 و x_2 و y_2 و x_3 و y_3 و e را که می‌توانند اعداد صحیح یا اعشاری باشند به ترتیب ذکر شده در خطوط جداگانه دریافت کند. در نظر بگیرید برای ورودی‌ها شرایط زیر موجودند.

$$0 \leq x_i, y_i \leq 100$$

$$0 \leq e \leq 10$$

توابع

در برنامه‌ی خود توابع زیر را تعریف و استفاده کنید:

- **تابع** $distance$**:** تابع مختصات دو نقطه‌ی حقیقی مثبت را در ورودی دریافت کرده و در خروجی فاصله‌ی آن دو نقطه از یکدیگر را بر می‌گرداند.
- **تابع** $compare$**:** تابع دو فاصله و ضریب سختی e را دریافت کرده و در صورت کمتر بودن نیروی انتخاب به کلاس رفتن رشته‌ی $Class$ و در صورت کمتر بودن نیروی انتخاب به خوابگاه رفتن رشته‌ی $Dorm$ را در خروجی بر می‌گرداند.

خروجی

خروجی برنامه‌ی شما بایستی در یک خط رشته‌ی $Class$ یا $Dorm$ را بر اساس اینکه کدام انتخاب *نیروی مصرفی دانشجو* را کمترین می‌کند چاپ کند.

مثال‌ها

ورودی نمونه ۱

```
1
2
3
6
6
10
0.5
```

خروجی نمونه ۱

```
Class
```

ورودی نمونه ۲

```
5
2
7
5
10
4
2.3
```

خروجی نمونه ۲

```
Dorm
```

Jafar's Goodbye Party

- محدودیت زمان: 10 ثانیه
- محدودیت حافظه: 512 مگابایت

داوری سوال به صورت دستی بوده و تست کیس ندارد.

آلیس برای جشن خداحافظی خارج رفتن جعفر می‌خواهد با جمعی از دوستانش به طور محرمانه برنامه‌ریزی کند به طوری که حتی اگر جعفر پیام‌هایشان را دید از نقشه‌شان بو نبرد. او برای این‌کار به دنبال ابزاری می‌گردد که برای رمز گذاری و رمز گشایی راحت پیام‌ها در اشتراک تمام گروه بگذارد و خوشبختانه یک برنامه‌ی رمز نامتقارن (الگوریتم رمزی که برای رمز گذاری و رمز گشایی پیامی از دو سری عدد متفاوت استفاده می‌کند) پیدا کرده است.

اما برنامه فقط با گرفتن پیام مورد نظر و یک کلید عمومی (اعدادی برای استفاده در الگوریتم رمز کردن پیام برای فردی که کلیدش را وارد می‌کنیم) یا کلید خصوصی (اعدادی برای استفاده در الگوریتم رمز گشایی پیامی که برای فردی با کلید عمومی او رمز شده است) به رمز گذاری یا رمز گشایی پیام می‌پردازد. یعنی هر فرد در گروه بایستی پیش از استفاده از برنامه برای خود کلید عمومی و خصوصی خاصی بسازد. به علت سختی محاسبات و کمبود دانش بعضی افراد گروه، آلیس از شما می‌خواهد که بر اساس روابط ریاضی ذکر شده در توضیحات برنامه، برنامه‌ای دیگر برای تولید کلیدها بسازید که افراد بدون نیاز به دانستن شیوه‌ی تولید کلیدها یا الگوریتم رمز، از برنامه‌ها به راحتی استفاده کنند.

بنابراین، نیاز است برنامه‌ای بنویسید که دو عدد صحیح p و q را در ورودی بگیرد و ابتدا بررسی کند که آیا اعداد ورودی اول هستند یا خیر؛ در صورت اول بودن، کلیدها را به کمک توابع تولید کند و سپس تک تک چاپ کند، در غیر این صورت $False$ را چاپ کند.

روابط ریاضی کلیدها

برای محاسبه‌ی مقادیر کلید عمومی و خصوصی، فرض کنید اعداد p و q دو عدد اول هستند و n عدد حاصل ضرب آن‌هاست. همچنین $\Phi(n)$ را به شکل زیر داریم:

$$\Phi(n) = (p-1) \times (q-1)$$

آنگاه کلید عمومی، اعداد n و e است که n همان حاصل ضرب دو عدد اول بوده و e نیز عددی تصادفی در بازه‌ی $(1, \Phi(n))$ می‌باشد که نسبت به $\Phi(n)$ اول است (یعنی بزرگ‌ترین مقسوم‌علیه مشترک عدد e و عدد $\Phi(n)$ یک می‌باشد). به بیانی دیگر:

$$n = p \times q$$

$$1 < e < \Phi(n) \quad gcd(e, \Phi(n)) = 1$$

همچنین کلید خصوصی، اعداد n و d است که d با رابطه‌ی زیر به کمک عدد تصادفی k تولید شده است:

$$d = \lfloor \frac{(k \times \Phi(n)) + 1}{e} \rfloor$$

بخش‌های برنامه

در نوشتن برنامه‌ی خود از 5 تابع، ماژول $math$ و ماژول $random$ برای اعمال متفاوت استفاده کنید. در ادامه، نکات پیاده‌سازی برنامه و توابع مورد نظر شرح داده شده‌اند.

ورودی

برنامه نیاز است دو عدد صحیح p و q را در دو ورودی در خطوط جدید دریافت کند. فرض کنید برای ورودی‌ها شرط زیر موجود است.

$$1 \leq p, q \leq 100$$

توابع

در برنامه‌ی خود توابع زیر را تعریف و استفاده کنید:

- **تابع $isPrime$:** تابع در ورودی عدد صحیحی دریافت کرده و در صورت اول بودن آن $True$ بر می‌گرداند، در غیر این صورت $False$ بر می‌گرداند.

- **تابع** $isRelativePrime$**:** تابع دو عدد صحیح در ورودی دریافت کرده و با محاسبه‌ی ب.م.م. اعداد، در صورت اول بودن آن‌ها نسبت به یکدیگر $True$ بر می‌گرداند، در غیر این صورت $False$ بر می‌گرداند.
- **تابع** $pickExponent$**:** تابع عدد صحیح $\Phi(n)$ را در ورودی دریافت کرده و تلاش بر تولید e تصادفی در بازه‌ی تعیین شده می‌کند. هرگاه e تصادفی با شرایط ذکر شده (اول بودن نسبت به $\Phi(n)$ ورودی، که به کمک تابع $isRelativePrime$ تشخیص داده می‌شود) یافت شد، آن را بر میگرداند.
- **تابع** $publicKey$**:** تابع سه عدد صحیح p و q و e را در ورودی دریافت کرده و بنابر رابطه‌ی ریاضی ذکر شده، کلید عمومی را برای اعداد تولید می‌کند و مقادیر n و e را بر می‌گرداند.
- **تابع** $privateKey$**:** تابع سه عدد صحیح p و q و e را در ورودی دریافت کرده و بنابر رابطه‌ی ریاضی ذکر شده، کلید خصوصی را برای اعداد تولید می‌کند و مقادیر n و d را بر می‌گرداند. در تولید عدد تصادفی k بازه‌ی عدد را شبیه به بازه‌ی e در نظر بگیرید و $seed$ را 17 قرار دهید.

خروجی

برنامه بایستی در صورت مناسب نبودن ورودی‌ها (اول نبودن آن‌ها) $False$ را چاپ کند. در غیر این صورت ابتدا کلید عمومی و سپس کلید خصوصی را تولید کرده و به ترتیب در دو خط چاپ کند، به طوری که اولین عدد هر خط مقدار n باشد.

مثال‌ها

در مثال‌ها دقت کنید که به دلیل تصادفی بودن تولید e، برای یک ورودی امکان تولید خروجی‌های متفاوت وجود دارد.

ورودی نمونه ۱

```
11
73
```

خروجی نمونه ۱

```
803 233
803 1656
```

ورودی نمونه ۲

```
11
73
```

خروجی نمونه ۲

```
803 451
803 855
```

ورودی نمونه ۳

```
23
4
```

خروجی نمونه ۳

```
False
```

ورودی نمونه ۴

```
23
13
```

خروجی نمونه ۴

```
299 155
299 364
```

ورودی نمونه ۵

```
24
12
```

خروجی نمونه ۵

```
False
```