

ثبت نام امن

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: آسان
- طراح: روزبه سلطانی

گروهی از دوستان شما تصمیم گرفته‌اند سیستمی طراحی کنند که در آن کاربران با وارد کردن ایمیل و رمزعبور خود ثبت نام کنند. برای جلوگیری از ثبت نام‌های نامعتبر، از شما خواسته شده برنامه‌ای بنویسید که بررسی کند آیا ورودی‌های کاربران معتبر هستند یا خیر.

شرایط اعتبارسنجی

ایمیل معتبر (Valid Email)

- نام کاربری (بخش قبل از @):
 - باید حداقل یک کاراکتر طول داشته باشد و با حروف انگلیسی یا یک رقم شروع شود.
 - فقط شامل حروف انگلیسی (کوچک و بزرگ)، ارقام و کاراکتر نقطه (.) باشد. (استفاده از کاراکترهایی مانند آندرلاین _ یا خط تیره - مجاز نیست).
- جداکننده:
 - شامل یک کاراکتر @ باشد.
- نام دامنه (بخش بین @ و .):
 - باید حداقل یک کاراکتر طول داشته باشد.
 - فقط شامل حروف انگلیسی و ارقام باشد. (استفاده از کاراکترهایی مثل نقطه یا خط تیره در این بخش مجاز نیست).
- پسوند دامنه (بخش بعد از نقطه آخر):
 - شامل یک کاراکتر نقطه (.) باشد.
 - بعد از نقطه، باید حداقل دو و حداکثر چهار کاراکتر حرف انگلیسی باشد (مانند ir ، com ، info).

مثال :

test123@mail.ir و user@example.com

رمزعبور معتبر (Valid Password)

- **طول:** حداقل ۸ کاراکتر طول داشته باشد.
- **حرف کوچک:** حداقل شامل یک حرف کوچک انگلیسی ([a-z]) باشد.
- **حرف بزرگ:** حداقل شامل یک حرف بزرگ انگلیسی ([A-Z]) باشد.
- **رقم:** حداقل شامل یک رقم باشد.
- **کاراکتر خاص:** حداقل شامل یک کاراکتر خاص از مجموعه [!@#\$%^&*()] باشد.

مثال :

Admin123# و P@ssw0rd!

ورودی

در خط اول یک عدد صحیح n داده می‌شود که نشان‌دهنده تعداد کاربران است. در n خط بعدی، در هر خط یک ایمیل و رمزعبور با فاصله از هم داده می‌شود.

$$1 \leq n \leq 100$$

$$1 \leq |Password|$$

$$|Email| \leq 100$$

خروجی

برای هر کاربر، در یک خط خروجی چاپ کنید:

- اگر هر دو معتبر بودند: Both Valid
- اگر فقط ایمیل معتبر بود: Email Valid
- اگر فقط رمزعبور معتبر بود: Password Valid
- اگر هیچ‌کدام معتبر نبودند: Invalid

مثال

ورودی نمونه ۱

3
user@gmail.com P@ssw0rd!
test@domain admin
wrongmail.com Pass123!

خروجی نمونه ۱

Both Valid
Invalid
Password Valid

ورودی نمونه ۲

2
a@b.c 123
hello@world.com Admin123#

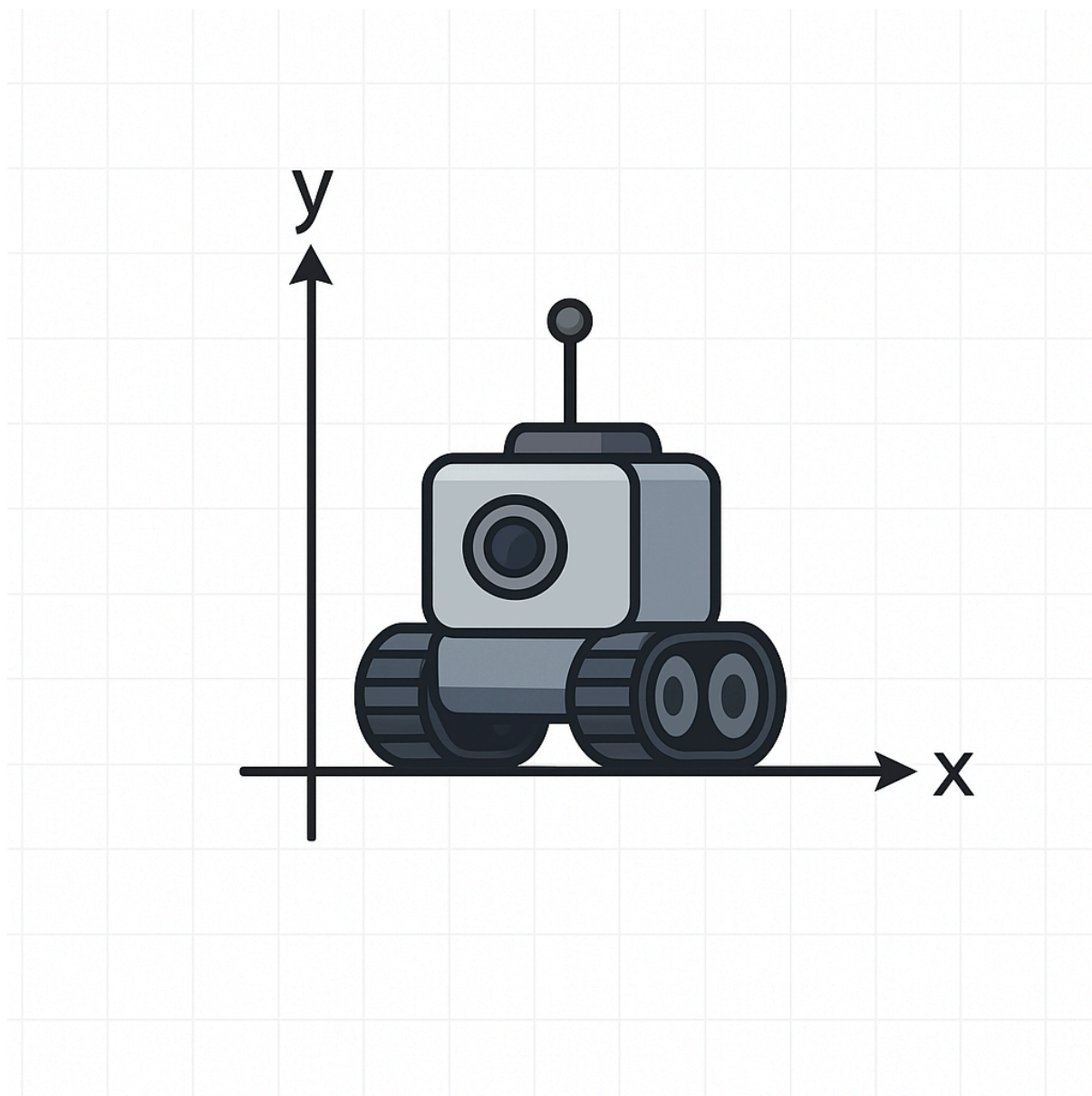
خروجی نمونه ۲

Invalid
Both Valid

Rotate and Go!

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: متوسط
- طراح: آریا زریاب

یک ربات ماشینی روی مبدا مختصات صفحه‌ی مختصات دو بعدی قرار دارد.
جهت این ماشین رو به $+\infty$ محور x ها است.



برای اینکه این ربات روی صفحه حرکت کند باید به آن فرمان بدهیم. هر فرمان دو حالت زیر را دارد:

فرمان Forward ، یعنی یک واحد به جلو برو.

فرمان Rotate ، یعنی ۹۰ درجه در خلاف جهت عقربه‌های ساعت در همان نقطه بچرخ.

قبل از حرکت ربات می‌توان به آن یک رشته از n فرمان داد و سپس این ربات به ترتیب این فرمان‌ها را از چپ به راست هر کدام را یکبار اجرا می‌کند.

آریا می‌خواهد مسیر حرکت ربات را مشخص کند. او یک مسیر را با دنباله‌ای از حرکت‌های U، R، L و D مشخص می‌کند که به ترتیب یعنی یک واحد به چپ (یا رفتن یک واحد به سمت x کمتر)، راست (یک واحد x بیشتر)، بالا (یک واحد y بیشتر) و پایین (یک واحد y کمتر) حرکت کن.

حال مسئله این است که اگر رشته‌ای از n کاراکتر که مسیر حرکت را به فرمتی که آریا ارائه می‌دهد به شما بدهند می‌توانید آن را به فرمتی که ربات حرکت می‌کند تبدیل کنید به طوری که دقیقاً همان مسیر مورد نظر آریا طی شود و کمترین تعداد عملیات انجام شود؟

ورودی

در سطر اول ورودی، عدد صحیح و مثبت n آمده که تعداد کاراکترهای رشته‌ی نشان دهنده‌ی مسیر مورد نظر آریا را نشان می‌دهد.

$$1 \leq n \leq 100$$

در سطر دوم ورودی، یک رشته از n کاراکتر L ، R ، U و D داده می‌شود که به ترتیب مسیر حرکت مورد نظر آریا را نشان می‌دهد.

خروجی

در سطر تنها سطر خروجی، یک رشته از حروف F و R چاپ کنید که نشان دهنده‌ی دنباله‌ی فرمان‌هایی است که به ربات داده می‌شود (منظور از F فرمان Forward و منظور از R فرمان Rotate است).

مثال

ورودی نمونه ۱

10

RRRUULDDDD

خروجی نمونه ۱

FFRFRFRFRFFF

ورودی نمونه ۲

4

UDRL

خروجی نمونه ۲

RFRRFRFRF

رمزنگاری سزار دوم

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: متوسط
- طراح: علیرضا متقی

ژولیوس سزار در تاریخ 60 قبل از میلاد مسیح، رمز انتقالی (Shift Cipher) را برای ارسال پیام های مهم جنگی در دوره پیش از امپراتوری روم ابداع کرد. این رمز هر حرف رو به t حرف جلوتر از خود تبدیل میکند و برای رمزگشایی متون، با استفاده از یک دستگاه تبدیل هر حرف به t حرف قبل خود را مشخص و به سرعت از روی آن متن اصلی را بازنویسی میکردند.



حال محسن و آریان دنبال راه دیگری برای پیچیدگی بیشتر نسبت به این روش قدیمی و منسوخ شده رمزنگاری هستند. آنها با تلاش برای زنده کردن مجدد سزار از او کمک میخواهند تا راه جدیدتری برای رمز کردن بسازد. سزار پس از برخاستن از قبر خود و کمی تأمل به یک الگوریتم جدید میرسد که به این صورت است:

ابتدا حرف آخر رشته را به اول رشته انتقال میدهیم و سپس تمامی حروف رشته را به حرف بعدی آن در الفبا تبدیل میکنیم. به عنوان مثال حرف c به d تبدیل میشود (همچنین حرف بعدی z در الفبا را a در نظر میگیریم). برای قویتر کردن رمزگذاری پیشنهادی، این عملیات k بار بر روی رشته انجام شود.

محسن و آریان از شما میخواهند که نتیجه متن رمز شده آنها را با این روش بدست بیاورید.

توجه کنید در هر مرحله از k بار عملیات، هر دوی «انتقال حرف از آخر به اول» و «تبدیل هر حرف به حرف بعدی» انجام می‌شود!

توجه: استفاده از *StringBuilder* در راه‌حل این سوال مجاز نیست.

ورودی

در خط اول ورودی n آمده که نشان دهنده‌ی طول پیام است.

در خط دوم ورودی k آمده که نشان دهنده‌ی تعداد باری است که عملیات رمزگزاری باید صورت گیرد.

در خط سوم ورودی رشته‌ی مورد نظر آمده، تضمین می‌شود تمام حروف آن از حروف کوچک انگلیسی است.

$$1 \leq n, k \leq 100$$

خروجی

در تنها خط خروجی رشته‌ی مورد نظر را پس از k بار رمزگزاری خروجی دهید.

مثال

ورودی نمونه ۱

3
1
abz

خروجی نمونه ۱

abc

یک مرحله رمزگزاری به صورت زیر انجام می‌شود:

- ابتدا حرف آخر رشته به اول آن می‌رود در نتیجه abz به zab تبدیل می‌شود.

- سپس تمامی حروف رشته با حرف بعدی الفبایی‌شان جابه‌جا می‌شوند در نتیجه zab به abc تبدیل می‌شود.

ورودی نمونه ۲

4
5
abcd

خروجی نمونه ۲

ifgh

دو عملیات اول روی رشته‌ی abcd به صورت زیر خواهد بود:

- عملیات اول:

◦ ابتدا حرف آخر رشته (حرف d) به اول رشته منتقل می‌شود و به رشته‌ی dabc خواهیم رسید.

◦ سپس تمامی حروف به حرف بعدی الفبا منتقل می‌شوند به رشته‌ی ebcd خواهیم رسید.

- عملیات دوم:

◦ ابتدا حرف آخر رشته (حرف d) به اول رشته منتقل می‌شود و به رشته‌ی debc خواهیم رسید.

◦ سپس تمامی حروف به حرف بعدی الفبا منتقل می‌شوند به رشته‌ی efcd خواهیم رسید.

۳ عملیات باقی‌مانده به طور مشابه صورت می‌گیرند و در انتها به رشته‌ی ifgh خواهیم رسید.

طلسم کتابخانه

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- سطح: سخت
- طراح: آناهیتا بیانی

در گوشه‌ای فراموش‌شده از کتابخانه‌ای کهن، میان گرد و غبار سده‌ها، کتابی را یافته‌اید که طلسمی در آن پنهان شده است، برای دست یافتن به قسمت مخفی کتابخانه باید واژه طلسم را بیابید.

اما طلسم این کتاب چند راز دارد:

۱. طلسم تنها زمانی قدرتمند است که **همهٔ حرف‌های لازم** را در خود داشته باشند.
۲. و هرگاه از حروف نفرین‌شده، **حتی یک‌بار** در آن دیده شود، راه ورود به قسمت مخفی برای همیشه بسته خواهد شد.
۳. **فاصله** بین **دو حرف** نباید از مقدار خاصی بیشتر باشد وگرنه قدرت خود را از دست خواهد داد.

خواسته مسئله:

از میان رشته‌ای از حروف (S)، کوتاه‌ترین بخش متوالی را بیابید که:

۱. **همهٔ حروف ضروری** (T) را داشته باشد. (رشته‌ی T ممکن است شامل حروف تکراری باشد، باید به تعداد لازم در زیررشته موجود باشند.)
۲. **هیچ‌کدام** از **حروف نفرین‌شده** (U) در آن نباشد.
۳. اگر چند زیررشته اندازه مشابهی داشتند، زیررشته‌ای را بیابید که از نظر **ترتیب الفبایی** (**lexicographical order**)، **جلوتر** از بقیه بیاید.
۴. در زیررشتهٔ طلسم، بین هر دو حرف مورد نیاز که پشت‌سر هم می‌آیند، نباید بیش از K کاراکتر **غیرضروری** وجود داشته باشد.

ورودی

خط اول شامل رشته‌ی S می‌باشد:

$$1 \leq |S| \leq 100$$

خط دوم شامل رشته‌ی T می‌باشد:

$$1 \leq |T| \leq 100$$

خط سوم شامل رشته‌ی U می‌باشد:

$$1 \leq |U| \leq 100$$

خط چهارم شامل عدد K می‌باشد:

$$1 \leq K \leq 100$$

خروجی

در صورت وجود زیررشته‌ای که شرایط طلسم را داشته باشد، **اندیس شروع و اندیس پایان آن** را در یک سطر چاپ کنید. اگر چنین زیررشته‌ای وجود نداشت، عدد **-1** را چاپ کنید.

مثال

ورودی نمونه ۱

```
ADBCODEBANC
ABC
DO
10
```

خروجی نمونه ۱

```
7 10
```

ورودی نمونه ۲

AXXXBXXXCXXBMMAC

ABC

M

2

خروجی نمونه ۲

-1