

فرار به بروجن

صدرا یک گنج به ارزش 10 میلیارد تومان در تهران پیدا کرده است. آریا که این خبر را فهمیده به دنبال صدرا افتاده تا گنج او را از او بدزدد. صدرا می داند اگر موفق شود به بروجن برسد از دست آریا در امان می ماند. فرض کنید ایران دارای n شهر است که تهران شهر t ام آن و بروجن شهر b ام آن است.

آریا گروهی از دوستان سده ای خود را اجیر کرده که در شهر های مختلف مستقر شوند تا صدرا را خفت کنند ولی صدرا از این موضوع با خبر است، بنابراین صدرا از پلیس کمک میخواد. پلیس می تواند در هر مرحله بازه ای از شهر های پشت سر هم به طول 2^k (k عدد حسابی است) را از سده ای ها پاکسازی کند.

پلیس ها وقت زیادی ندارند. برای همین از شما می خواهند کمترین تعداد عملیات برای پاکسازی مسیر بین بروجن و تهران را بگویند.

ورودی

در سطر اول عدد n آمده که نشان دهنده تعداد شهر های ایران است.

در سطر دوم یک رشته به طول n آمده است. شهر هایی که در آن سده ای وجود دارد حرف S و بقیه خانه ها حرف P هستند. تضمین می شود که در خانه های b و t سده ای وجود ندارد.

در سطر سوم b و t به ترتیب آمده اند.

$$1 \leq n \leq 1000$$

$$1 \leq b, t \leq n$$

خروجی

در تنها سطر خروجی، کمترین تعداد عملیات برای پاکسازی مسیر صدرا از سده ای ها را بگویند.

مثال ها

ورودی نمونه 1

10
PSPSSSPSP
1 9

خروجی نمونه 1

4

بروجن در خانه ی 1 و تهران در خانه ی 9 قرار دارد. در این مسیر ابتدا در خانه ی 2 یک سده ای وجود دارد که نیازمند یک عملیات (2^0) است. سپس در خانه ی 4 و 5 و 6 سده ای ها مستقر شدند که برای پاکسازی نیازمند 2 مرحله است ($2^1 + 2^0$). در آخر هم در خانه ی 6 یک سده ای وجود دارد که با 1 مرحله عملیات پاکسازی می شود. پس نهایتا کمترین تعداد عملیات ها 4 تا است.

ورودی نمونه 2

9
SPPSSPSPS
8 3

خروجی نمونه 2

2

در مسیر خانه هشتم به سوم تنها در خانه های 4 و 5 و 7 سده ای وجود دارد که پلیس ها طی یک مرحله زورگیر خانه ی 4 و 5 و در مرحله ی بعد زورگیر خانه ی 7 را دستگیر می کنند. در حرکت اول یک بازه به طول 2 و در حرکت دوم یک بازه به طول 1 پاکسازی شد که طول هر دو بازه توانی از 2 بود.

پ.ن: سده شهری در استان اصفهان است.

بروجن هم شهری بسیار زیبا در استان چهارمحال و بختیاری است.

تحلیل مدار گلابی

تشریح کلی سوال

در این سوال قصد داریم برنامه ای بنویسیم که ورودی آن توصیف یک مدار و خروجی آن، یک تحلیل عددی ساده از آن مدار باشد. المان های مدار فقط شامل منبع ولتاژ و مقاومت است. همچنین، تنها دو نوع مدار داریم:

• 1 - مدار سری

• 2 - مدار موازی

در مدار سری، تا حداکثر 4 مقاومت سری، با منبع ولتاژ سری می شوند. در مدار موازی، تا حداکثر 4 مقاومت موازی، با منبع ولتاژ موازی می شوند.

در خروجی برنامه، بایستی مقاومت معادل دیده شده از دو سر منبع و توان اتلافی مدار بدست آیند.

ورودی های برنامه شامل پارامتر های زیر است:

- ولتاژ منبع = V (ولت)
- نوع مدار (سری یا موازی) = $type$
- تعداد مقاومت هایی که می خواهیم در مدار قرار دهیم = Num
- میزان مقاومت آی ام که در هر خط به طور جداگانه دریافت می شود = $R[i]$ (اهم)

تضمین می شود که ورودی های مدار بیش از 2 رقم اعشار نشود.

رنج هر پارامتر ورودی عبارت است از:

$$0 \leq V \leq 100$$

$$1 \leq type \leq 2$$

$$1 \leq Num \leq 4$$

$$0 \leq R[i] \leq 1000000$$

اگر type=1 باشد به معنی سری بودن مدار و در صورتی که type=2 باشد به معنی موازی بودن مدار است.

در خروجی نیز میزان مقاومت معادل دیده شده از دو سر منبع و میزان توان اتلافی مدار، به ترتیب با واحد کیلو اهم و میلی وات گزارش می شود که **بایستی هر دو حداکثر به دو رقم اعشار گرد و سپس نمایش داده شود.** (برای گرد کردن پیشنهاد می شود از DecimalFormat استفاده کنید. در مورد آن جستجو کنید.)

دقت کنید که اگر ورودی نوع مدار اشتباه باشد، (به عنوان مثال type = 10) بایستی پیغام Invalid Input نمایش داده شود. همچنین اگر در مدار اتصال کوتاه رخ داد، بجای محاسبه توان باید پیغام Short Circuit نمایش داده شود.

برای وضوح بیشتر سوال، به مثال های زیر دقت کنید:

مثال

ورودی نمونه 1

5
1
2
1500
3500

در این مثال در ورودی خط اول ابتدا ولتاژ منبع تغذیه داده شده است که 5 ولت می باشد. ($V=5$) در خط دوم نوع مدار انتخاب شده است. (type=1) در خط سوم تعداد مقاومت هایی است که به صورت سری با منبع ولتاژ در مدار قرار می گیرند که در اینجا این مقدار 2 می باشد. (Num) در خط چهارم و پنجم، به ترتیب میزان مقاومت اول و دوم با واحد اهم وارد شده اند. ($R[i]$)

خروجی نمونه 1

Total Res: 5 Kohm

Total Pow: 5 mW

در خروجی نیز ابتدا در خط اول میزان مقاومت معادل دیده شده از دو سر منبع با واحد کیلو اهم و در خط دوم میزان توان اتلافی مدار با واحد میلی وات آورده می شود. (به فواصل بین کلمات و case sensitivity دقت کنید)

ورودی نمونه 2

10

1

2

2598.87

2340.1

خروجی نمونه 2

Total Res: 4.94 Kohm

Total Pow: 20.25 mW

ورودی نمونه 3

27.09

0

خروجی نمونه 3

Invalid Input

ورودی نمونه 4

100
2
4
1000
10000
100000.25
1000000

خروجی نمونه 4

Total Res: 0.9 Kohm
Total Pow: 11110 mW

ورودی نمونه 5

0
2
4
21
1
0
14000

خروجی نمونه 5

Total Res: 0 Kohm
Short Circuit

به راحتی سودوکو

دانشگاه در کنار برگزاری مسابقات انتگرال گیری قصد دارد مسابقات حل سودوکو سرعتی را برگزار کند. عمو مصطفی دوست دارد در این مسابقات شرکت کند اما به دلیل ددلاین های زیاد وقت شرکت کردن را ندارد. به همین خاطر از شما می خواهد تا یک حل کننده سودوکو برایش طراحی کنید.

در این مسئله شما یک حل کننده سودوکو می نویسید به طوری که یک سودوکو ۶ در ۶ را حل کند. برای حل یک جدول سودوکو ۶ در ۶ باید ۳ نکته را رعایت کرد.

۱- در هر سطر اعداد باید غیر تکراری باشند.

۲- در هر ستون اعداد باید غیر تکراری باشند.

۳- در هر ۶ خانه شامل ۲ سطر و ۳ ستون اعداد باید غیر تکراری باشند.

در جدول زیر نمونه ای از جدول سودوکو ۶*۶ داده شده است:

| ۶ ۴ ۵ || ۱ ۲ ۳ |

| ۲ ۳ ۱ || ۶ ۴ ۵ |

| ۵ ۶ ۲ || ۴ ۳ ۱ |

| ۴ ۱ ۳ || ۵ ۶ ۲ |

| ۳ ۵ ۶ || ۲ ۱ ۴ |

| ۱ ۲ ۴ || ۳ ۵ ۶ |

می توانید از ArrayList نیز استفاده کنید. لینک یادگیری :

<https://maktabkhooneh.org/mag/arraylist-in-java/>

ورودی

در ورودی در ۶ خط ۶ عدد از ۱ تا ۶ با جاهای خالی x داده می شود. بین هر عدد ورودی یک فاصله گذاشته است.

تضمین می شود که در هر سطر یا هر ستون حداکثر ۲ جای خالی x وجود دارد.

نکته: ورودی را حتما با `nextline` بگیرید.

خروجی

خروجی برنامه جدول سودوکو کامل شده است.

مثال

ورودی نمونه ۱

```
x 2 1 5 3 6
5 6 x 2 1 4
3 4 2 1 6 x
1 5 6 x 4 2
2 x 4 6 5 3
6 3 5 4 x 1
```

خروجی نمونه ۱

```
4 2 1 5 3 6
5 6 3 2 1 4
3 4 2 1 6 5
1 5 6 3 4 2
2 1 4 6 5 3
6 3 5 4 2 1
```

ورودی نمونه ۲

3 2 x 4 x 6
1 x 6 3 x 5
5 6 3 x 4 x
2 x x 5 6 3
x 3 2 x 5 1
x 5 1 2 3 x

خروجی نمونه ۲

3 2 5 4 1 6
1 4 6 3 2 5
5 6 3 1 4 2
2 1 4 5 6 3
4 3 2 6 5 1
6 5 1 2 3 4

هزارتو (امتیازی)

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

در یک سرزمین دور، یک جواهر افسانه‌ای وجود دارد که در حال حاضر گم شده است. می‌دانیم که این جواهر در یک هزارتو قرار دارد، و شما قرار است به کمک توانایی های برنامه نویسی خود، با پیمایش این هزارتو، جواهر را بیابید. هزارتو به صورت یک آرایه‌ی دو بعدی نمایش داده می‌شود که در آن:

- 0 نشان دهنده‌ی فضای خالی است که می‌تواند پیمایش شود.
- 1 نشان دهنده‌ی مانع است که قابل پیمایش نیست.
- 9 نشان دهنده‌ی محل قرارگیری جواهر است.

پیمایش از درایه‌ی اول هزارتو آغاز می‌شود و حرکت در چهار جهت بالا، پایین، راست و چپ مجاز است. برنامه‌ی شما باید ابعاد هزارتو و آرایه‌ی آن را به عنوان ورودی دریافت کند و در خروجی کمترین تعداد حرکات برای رسیدن به جواهر را اعلام کند.

ورودی

در سطر اول ورودی، تعداد سطرها و ستون‌های آرایه‌ی هزارتو (به ترتیب) داده می‌شود. سپس آرایه‌ی هزارتو سطر به سطر داده می‌شود.

خروجی

در صورتی که مسیری به جواهر وجود داشته باشد، در خروجی کمترین تعداد حرکات برای رسیدن به جواهر و در غیر این صورت عدد 0 باید چاپ شود.

مثال

ورودی نمونه ۱

```

4 4
0 1 0 0
0 1 0 1
0 0 0 0
0 1 1 9

```

خروجی نمونه ۱

```

6

```

ورودی نمونه ۲

```

3 3
0 1 0
1 1 0
0 1 9

```

خروجی نمونه ۲

```

0

```

ورودی نمونه ۳

```

5 6
0 0 0 0 0 1
0 1 1 1 0 1
0 0 1 0 0 1
0 0 0 0 1 1
0 1 1 0 0 9

```

خروجی نمونه ۳

9