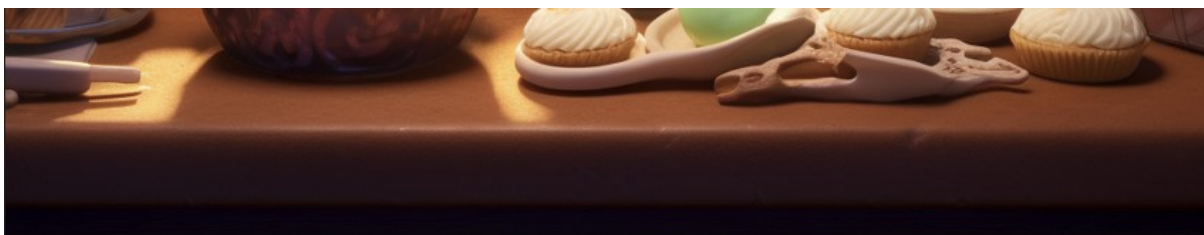


## بحران برق

- محدودیت زمان: ۰.۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

کارخانه‌ی هیولاها اخیراً دچار بحران شده، برق جیره‌بندی شده و کپسول‌های جیغ به طور سهمیه‌ای در هر منطقه پخش می‌شوند. به همین دلیل سالیوان، هیولای آبی ما، مسئولیت دسته‌بندی کپسول‌ها برای ارسال به مناطق را به عهده گرفته. هر منطقه کدی منحصر به فرد دارد که متشکل از تعدادی بایت است. سالیوان که باید روی هر کپسول یک کد منطقه درج می‌کرد، تصمیم گرفت برای خوانایی بهتر به جای یک دنباله‌ی بایتی، یک دنباله‌ی عددی درج کند. اما بعداً معلوم شد که مسئول قبلی، روش دیگری را برای درج کدها به پیش گرفته بود. سالیوان به پیش شما آمده و درخواست کرده با توجه به توضیحاتی که در ادامه می‌آید، برنامه‌ای بنویسید که به او در تشخیص هم‌منطقه‌ای بودن دو کپسول جیغ کمک کند.





می‌دانید که در کامپیوترها تمام داده‌ها در قالب 0 و 1 ذخیره می‌شوند؛ و امروزه زبان C عموماً یک char را در قالب 1 بایت و یک int را در قالب 4 بایت ذخیره می‌کند.

با توجه به این، فرض کنید به شما یک int داده شده است. شما می‌توانید به آن به چشم 4 بایت نگاه کنید، سپس به هر بایت به چشم یک char نگاه کنید. یعنی می‌شود گفت که هر int معادل با چهار char است. درواقع اگر 4 بایت data (داده) داشته باشید و کسی به شما نگوید که آن 4 بایت مربوط به چه تایپی از data است، شما نمی‌توانید بفهمید که آیا با یک int سر و کار دارید یا با چهار char یا با چیزی دیگر.

برای مثال فرض کنید اگر در حافظه‌ی یک کامپیوتر عدد 65 را (در قالب int) ذخیره کنیم، به شکل 4 بایت زیر در می‌آید (هر بایت را با 8 بیت نشان خواهیم داد):

```
00000000 00000000 00000000 01000001
```

البته این عدد در اکثر سیستم‌های امروزی به ترتیب دیگری ذخیره می‌شود که در مسائل بعدی بیشتر با آن آشنا خواهید شد، اما فعلاً از آن صرف نظر کنید.

حال اگر شخصی بخواهد این بایت‌ها را به شکل char بخواند، به ترتیب کاراکترهای زیر را خواهد خواند:

```
NUL NUL NUL A
```

با رجوع به جدول ASCII می‌بینید که کد 0 متعلق به کاراکتر NUL و کد 65 متعلق به کاراکتر A است.

با این اوصاف، از شما خواسته شده تا برنامه‌ای بنویسید که دنباله‌ای از اعداد و حروف را ورودی بگیرد و دو دنباله را با هم مقایسه کند و اگر دنباله‌ی بیتی یکسانی برای ذخیره آن‌ها به کار می‌رود عبارت Yes وگرنه عبارت No را چاپ کند. اعضای هر دنباله در حافظه پشت‌سرهم ذخیره می‌شوند. (شما نحوه‌ی

ذخیره شدن یک `int` در حافظه را نمی‌دانید و صرفاً می‌دانید که 4 بایت در حافظه اشغال می‌کند.)

## ورودی

در خط اول به شما عدد  $n$  داده می‌شود. در هر یک از  $n$  خط بعدی به شما یک عدد داده می‌شود که باید تایپ آن را `int` در نظر بگیرید. سپس در خط آخر به شما  $4n$  حرف انگلیسی بزرگ یا کوچک داده می‌شود.

$$1 \leq n \leq 10^6$$

## خروجی

خروجی برنامه یک خط شامل عبارت `Yes` یا `No` می‌باشد.

## مثال

### ورودی نمونه ۱

```
1
1684234849
abcd
```

### خروجی نمونه ۱

```
Yes
```

### ورودی نمونه ۲

```
3
1752790119
1096895601
1346795639
gtyhqLaAwxFP
```

خروجی نمونه ۲

Yes

ورودی نمونه 3

2  
1802466674  
1886609765  
ruokEASP

خروجی نمونه 3

No