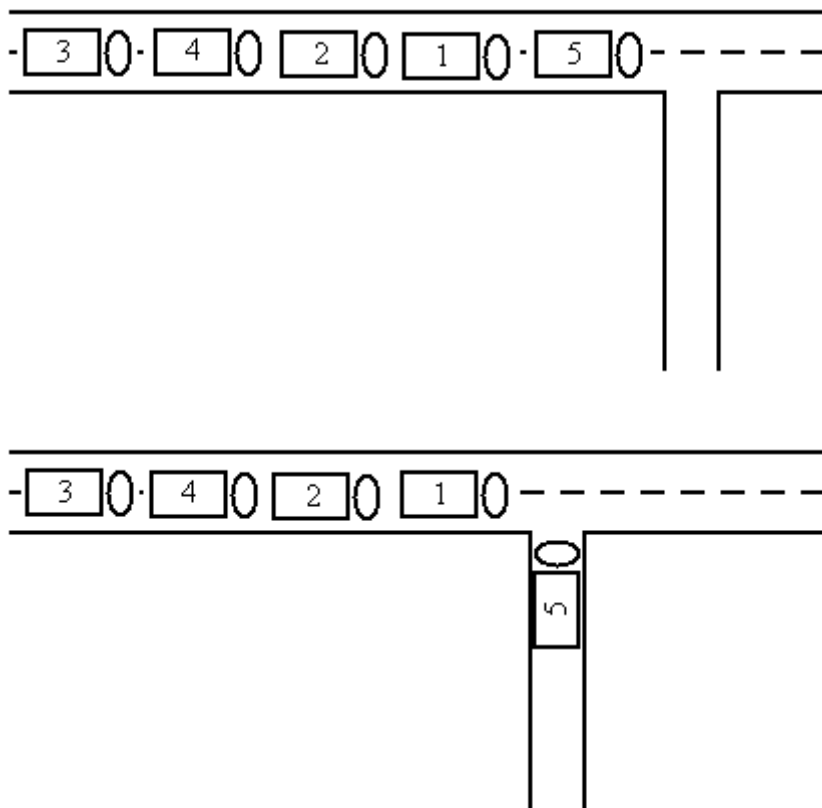


رژه خیابانی

ارتش بولیوی پس از پیروزی مجدد سوسیالیست‌ها در انتخابات ریاست جمهوری یک مانور ترتیب داده‌است ولی به دلیل قصور یکی از مقامات کامیون‌های ارتش بدون ترتیب مدنظر فرماندهان وارد خیابان شده‌اند. روی هر کامیون یک عدد بین یک تا N نوشته شده است (N تعداد کامیون‌هاست و هیچ دو کامیونی عدد یکسان ندارند) و فرماندهان قصد دارند برای زیبایی بیشتر کامیون‌ها به ترتیب عدد نوشته شده در خیابان حرکت کنند ولی به اشتباه کامیون‌ها بدون ترتیب حرکت کرده‌اند و یک خیابان قبل از رسیدن به محل مانور متوجه این موضوع می‌شوند و از شما می‌خواهند که کمک کنید تا ترتیب کامیون‌ها درست شود.

خیابان‌ها تنها به اندازه‌ی یک کامیون پهنا دارند و امکان سبقت گرفتن نیست اما خوشبختانه قبل از رسیدن به محل اصلی یک خیابان فرعی وجود دارد که میتوان از آن برای جابه‌جا کردن ترتیب کامیون‌ها استفاده کرد. برای درک بهتر ابتدا به شکل‌های زیر توجه کنید:



همانطور که مشاهده می‌شود هر کامیون می‌تواند به جلو رفته و سپس دنده عقب وارد خیابان فرعی بشود و منتظر حرکت ماشین‌های عقب‌تر از خود بماند تا زمانی که نوبتش می‌شود بتواند حرکت کند. برای مثال در شکل بالا ابتدا کامیون ۵ وارد فرعی می‌شود سپس ۲ و ۱ به محل مانور می‌روند پس از آن ۴ هم وارد فرعی می‌شود و جلوی ۵ قرار می‌گیرد و پس از حرکت کامیون ۳ کامیون‌های ۴ و ۵ هم به ترتیب وارد محل مانور می‌شوند و به این صورت کامیون‌ها طبق نقشه‌ی مدنظر فرماندهان به ترتیب عدد وارد می‌شوند و خطر آبروریزی حزب رفع می‌شود.

سوال به این شکل است که شما با دریافت عدد روی کامیون‌ها باید با چاپ ۱ یا ۰ بگویید که آیا می‌توان کامیون‌ها را به ترتیب وارد محل مانور کرد یا خیر.

ورودی

در تنها خط ورودی شماره‌ی کامیون‌ها به ترتیب قرار گرفتن در خیابان داده می‌شود. (دقت کنید که اولین شماره از سمت چپ جلوترین ماشین است.)

$$N \leq 26000$$

خروجی

اگر می‌توان طبق روش مشخص شده کامیون‌ها را به ترتیب وارد محل مانور کرد yes و در غیر اینصورت no چاپ کنید.

مثال

ورودی نمونه ۱

5 1 2 3 4

خروجی نمونه ۱

yes

توضیحات ورودی:

نکته‌ی مهم این است که ترتیب نوشتن اعداد از جلوترین ماشین است یعنی در ورودی ۱ عدد اول که ۵ است شماره‌ی جلوترین و اولین کامیون موجود در خیابان است. این ۵ عدد دقیقا اعداد مطرح شده در صورت سوال هستند که طبق روش توضیح داده شده می‌توانند به راحتی مرتب شده و وارد محل مانور شوند.

ورودی نمونه ۲

1 3 4 2 5

خروجی نمونه ۲

no

توضیحات ورودی:

این ۵ کامیون به هیچ شکلی امکان ندارد بتوانند مرتب شده و وارد محل مانور شوند زیرا اگر کامیون شماره‌ی ۱ که جلوترین است برود سپس ۳ و ۴ باید وارد خیابان فرعی شوند تا ۲ بتواند حرکت کند اول از آنجایی که ۳ جلوتر از ۴ است در خیابان فرعی ۴ جلوتر از ۳ قرار می‌گیرد در نتیجه به هیچ وجه امکان ندارد که دیگر کامیون ۳ بتواند جلوتر از ۴ حرکت کند.

پیچیدگی زمانی حل این مسئله از $O(n)$ می‌باشد

وقفه‌های با اولویت

می‌خواهیم سیستم وقفه‌ها در کامپیوتر را به شکل ساده شبیه‌سازی کنیم به این شکل که برنامه‌ی ما دو نوع دستور دارد. دستور اول به شکل int نوشته می‌شود که در آن n شماره ی وقفه و t مدت زمان مورد نیاز برای مدیریت وقفه است. دستور دوم هم به شکل tn نوشته می‌شود که در آن n مدت زمانی است که گذشته‌است. عملکرد برنامه به این صورت است که همواره آخرین وقفه‌ای که وارد شده‌است پردازش می‌شود و با گرفتن دستور int وقفه‌ی جدید در اولویت قرار می‌گیرد و اگر وقفه‌ی دیگر در حال پردازش بود به کنار می‌رود و برنامه به سراغ این وقفه می‌رود. (همواره آخرین وقفه پردازش می‌شود) حال با دریافت دستور tn برنامه به مدت n ثانیه به پردازش وقفه‌ها مشغول می‌شود و به ترتیبی که گفته شد هر وقفه که پردازش شد از بین می‌رود و وقفه‌ی بعدی مورد پردازش قرار می‌گیرد. با هربار دریافت دستور tn برنامه‌ی شما باید وقفه‌ای که پس از اتمام این زمان در حال پردازش است را چاپ کند.

ورودی

در خط اول عدد N که تعداد دستورات است و سپس در هر یک از N خط بعدی یکی از دستورات int یا tn داده می‌شود.

$$n \leq 100000$$

خروجی

هربار که tn در ورودی داده می‌شود برنامه باید به اندازه زمان t مشغول به پردازش وقفه‌ها شود و نهایتاً وقفه‌ای که در پایان این زمان در حال پردازش است را چاپ کند. اگر وقفه‌ای موجود نبود باید main چاپ شود.

مثال

ورودی نمونه ۱

```

8
i1 5
i2 6
t 5
t 1
i3 4
t 0
t 6
t 10

```

خروجی نمونه ۱

```

i2
i1
i3
i1
main

```

توضیحات ورودی: در ابتدا وقفه‌ی شماره‌ی ۱ که به ۵ ثانیه زمان نیاز دارد در اولویت پردازش قرار می‌گیرد سپس وقفه‌ی دو وارد می‌شود و چون جدیدتر است در اولویت اول قرار می‌گیرد. حال با دریافت t 5 برنامه ۵ ثانیه و مشغول به پردازش وقفه‌ها می‌شود که چون آخرین وقفه شماره‌ی ۲ است از ۶ ثانیه‌ی مورد نیاز این وقفه ۵ ثانیه پردازش می‌شود. پس در این مرحله i2 در مرحله‌ی پردازش است که باید در خط اول خروجی i2 چاپ شود. سپس ۱ ثانیه‌ی دیگر برنامه به پردازش مشغول می‌شود که در اینجا وقفه‌ی شماره‌ی ۲ هر ۶ ثانیه‌اش تکمیل می‌شود و خارج می‌شود در نتیجه پس از پایان این ۱ ثانیه وقفه‌ی شماره‌ی ۱ در اولویت پردازش قرار دارد. در خط پنجم وقفه‌ی شماره‌ی ۳ داده می‌شود که چون آخرین وقفه‌ی وارد شده‌است در اولویت اول قرار می‌گیرد پس از آن ۰ ثانیه زمان پردازش مصرف می‌شود که تغییری در زمان باقی‌مانده‌ی وقفه‌ها ایجاد نمی‌کند و پس از پایان این صفر ثانیه همچنان وقفه‌ی سوم در حال پردازش که به همین خاطر در خط سوم وقفه‌ی ۳ چاپ می‌شود پس از این صفر ثانیه ۶ ثانیه زمان پردازش داده‌میشود که این ۶ ثانیه باعث می‌شود وقفه‌ی شماره ۳ که به ۴ ثانیه زمان نیاز داشت به شکل کامل پردازش و خارج شود و ۲ ثانیه هم از ۵ ثان که در نتیجه پس از پایان این زمان وقفه‌ی ۱ باقی‌ماند که همان هم چاپ می‌شود در آخر هم ۱۰ ثانیه زمان داده‌شده که وقفه‌ی ۱ تمام می‌شود با این زمان و در پایان آن چون دیگر وقفه‌ای در حال پردازش نیست عبارت main چاپ می‌شود.

پیچیدگی زمانی حل این مسئله از $O(n)$ می باشد

رمزهای حلقوی

احمد از دایره‌های چرخان خوشش می‌آید او میخواهد قفلی اختراع کند که کسی نتواند آن را باز کند مگر آنکه ساختمان داده‌ها را بلد باشد

قفل متشکل از n گوی مشکی چرخان است که روی هر کدام از آنها k عدد نوشته شده، برای باز کردن قفل یک گوی راهنمای قرمز چرخان نیز وجود دارد! که روی آن h عدد نوشته شده

در هر مرحله با توجه به راهنمای قفل می‌توانیم اولین گوی را برداریم hi تا جا به جا کنیم (اگر hi مثبت باشد گوی را به جلو و اگر منفی باشد گوی را به عقب می‌چرخانیم) ولی باید گوی را در آخر صف قرار دهیم

ورودی

در ابتدا دو عدد n, k که تعداد گوی‌های مشکی و تعداد اعداد روی هر گوی است می‌آید و در n خط بعدی، اعداد روی گوی i ام می‌آید. سپس در یک خط عدد h که تعداد اعداد روی گوی قرمز است و در l خط بعدی اعداد روی گوی قرمز می‌آید در نهایت عدد t که برابر با تعداد دفعاتی که باید دستورالعمل‌ها را بخوانید تا به رمز برسید می‌آید.

$$1 \leq n, k \leq 10000$$

$$-10000000 \leq hi \leq 10000000$$

خروجی

شما باید رمز را چاپ کنید

مثال

ورودی نمونه ۱

```

4 5
3 8 4 9 0
2 5 7 3 5
8 5 6 4 5
2 3 6 9 4
3
1 -2 -1
4

```

خروجی نمونه ۱

```

8353

```

توضیحات:

بعد از ۴ بار خواندن دستورالعمل ها گوی ها به شکل زیر هستند که رمز را می سازند

```

8 4 9 0 3

```

```

3 5 2 5 7

```

```

5 8 5 6 4

```

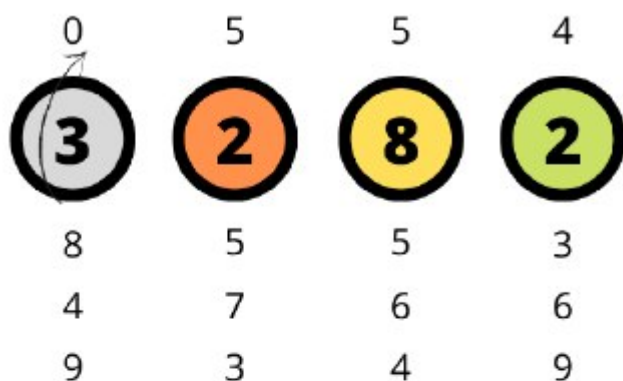
```

3 6 9 4 2

```

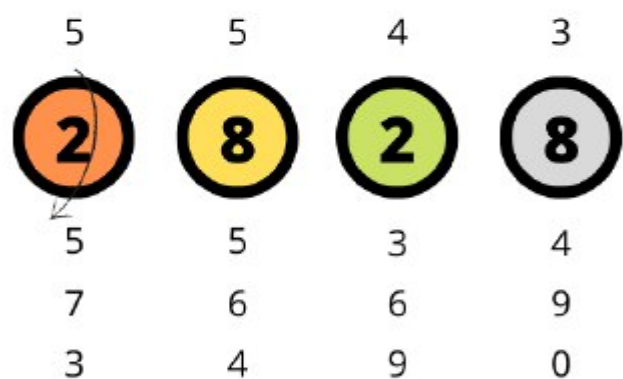
نکات: 1- محدودیت زمانی این مسئله از $O((t+n)^k)$ می باشد. 2- ممکن است اعداد روی گوی قرمز بزرگتر از k باشند.

برای توضیحات بیشتر به تصویر زیر توجه کنید:



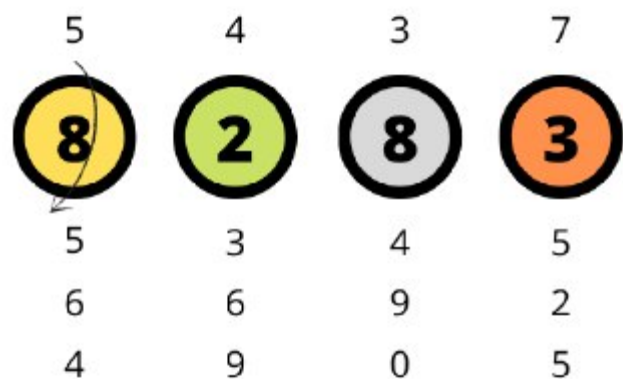
1

اولین توپ را یکی به جلو میچرخانیم و آخر صف قرار می‌دهیم



-2

اولین توپ را ۲ تا به عقب میچرخانیم و آخر صف قرار می‌دهیم











-1

اولین توپ را یکی به عقب میچرخانیم و آخر صف قرار می‌دهیم



1

			
3	4	5	8
6	9	2	5
9	0	5	6
3	7	4	2
			
4	5	8	6
9	2	5	9
0	5	6	4

توپ قرمز (دستورالعمل)
۳ بار خواند شده برای
همین بار چهارم دوباره
عدد ۱+ را نشان می‌دهد

تمام دستورالعمل‌ها
اجرا شده و رمز برابر
۸۳۵۳ است

محموله‌ی سری

ارتش بولیوی پس از مانور موفقیت آمیز خود تصمیم گرفت از ایده‌ی جابه‌جایی کامیون‌ها استفاده کند. از آنجایی که بولیوی بیش از ۲۲ درصد از ذخایر لیتیوم جهان را در اختیار دارد مسئله‌ی لو رفتن محموله‌ی کامیون‌های حامل لیتیوم برای این کشور بسیار حیاتی است. محموله‌ها به این شکل هستند که در هربار خروج کامیون‌ها از معدن برخی از آن‌ها حامل لیتیوم بوده و برخی نیستند به همین خاطر در اینجا شما باید لیست کامیون‌ها را به ترتیب خروج از معدن گرفته(هر کامیون دارای یک شماره است و برخی کامیون‌ها ممکن است شماره‌ی یکسان داشته باشند.) و طبق الگوریتم خواسته شده لیست را نامرتب کنید و اعداد با ترتیب جدید را خروجی دهید تا احتمالاً لو رفتن کامیون‌های حامل لیتیوم کمتر شود. الگوریتم نامرتب‌سازی شماره‌ها به این شکل است که آخرین کامیون به اول صف آمده و سپس یکی درمیان کامیون‌ها از آخر بین کامیون‌های اول صف می‌آیند. برای مثال اگر ورودی ما:

1 2 3 4

باشد پس از اعمال الگوریتم باید ترتیب به این شکل بشود:

4 1 3 2

یعنی آخرین کامیون می‌آید اول و به جای اولین کامیون قرار می‌گیرد، پس از آن کامیون یکی مانده به آخر بین کامیون اول و دوم می‌آید و در آخر هم کامیون دوم.

یا مثلاً:

1 2 3 4 5 6 7

تبدیل می‌شود به:

7 1 6 2 5 3 4

ورودی

در تنها خط ورودی شماره‌ی کامیون‌ها به ترتیب خروج از معدن داده‌میشود.

n = تعداد ورودی

m_i = مقدار هرکدام از ورودی‌ها

$$1 \leq n, m \leq 1000000$$

خروجی

در یک خط شماره‌ی کامیون‌ها را پس از نامرتب‌سازی نمایش دهید.

مثال

ورودی نمونه ۱

1 5 3 2 9 7 6 4

خروجی نمونه ۱

4 1 6 5 7 3 9 2

نکته: محدودیت زمانی این سوال از $O(n)$ است.

پشته‌های وسط‌دار

پشته ای پیاده سازی کنید که قابلیت اجرای دستورات زیر را داشته باشد و با گرفتن هر کدام از دستورات عملیات خواسته شده را روی پشته انجام داده و خروجی مطلوب را چاپ کند.

دستورات به شرح زیر می‌باشند:

دستور **push k**: عدد k به پشته اضافه شود.

دستور **pop**: آخرین عدد **push** شده در پشته حذف شود.

دستور **print**: اعداد موجود در پشته به ترتیب از آخرین عضو **push** شده تا اولین عضو **push** شده چاپ شوند.

دستور **findMiddle**: عنصر وسط پشته باید چاپ شود. (در صورت زوج بودن تعداد، عنصری که زودتر اضافه شده‌است چاپ می‌شود.)

دستور **removeMiddle**: عنصر وسط پشته حذف شود.

دستور **finish**: گرفتن ورودی‌ها به اتمام میرسد و برنامه تمام می‌شود.

نکته: اگر لیست خالی بود هنگام ورود دستور **findMiddle** باید 1- چاپ شود و هنگام آمدن دستورهای **print** باید خط خالی چاپ شود هنگام آمدن دستور **removeMiddle** و **pop** هم نباید اتفاقی بیفتد.

نکته: تمامی دستورات به جز **print**، از $O(1)$ هستند. دستور **print** از $O(n)$ می‌باشد.

ورودی

در هر خط یکی از دستورات گفته شده داده می‌شود (تعداد خطوط ورودی $n =$)

$$n < 10^6$$

دستور **push k**:

$$0 \leq k \leq 10^6$$

خروجی

باید هرکدام از خروجی‌های مطلوب را به ترتیب در خطوط جدا چاپ کنید.

مثال

ورودی نمونه ۱

```
push 1
push 2
push 3
push 4
print
findMiddle
removeMiddle
print
findMiddle
finish
```

خروجی نمونه ۱

```
4 3 2 1
2
4 3 1
3
```

ورودی نمونه ۲

```
push 10
push 20
pop
print
```

```
findMiddle  
finish
```

خروجی نمونه ۲

```
10  
10
```