



به نام خداوند بخشنده مهربان

تمرین کامپیوتری شماره ۳

محمد فغان پور گنجی

mfg1376@gmail.com

آذر ماه ۱۳۹۶

سیستم عامل

فرزاد خیلی خفن شده و تصمیم گرفته که سیستم عامل جدیدی برای خودش بسازد. برای پیاده سازی مدل دایرکتوری ها، می خواهد از روش لینوکس استفاده کند. برای پیاده سازی دستور های *cd* و *pwd* از شما کمک خواسته است. دستور اول، یک آرگومان (دایرکتوری مورد نظر) را می گیرد و به آن دایرکتوری می رود، دستور دوم هم دایرکتوری فعلی را چاپ می کند.

ساختار دایرکتوری ها به شکل یک درخت است که از یک ریشه (*root*) شروع می شود، که با کاراکتر / نشان داده می شود، بقیه دایرکتوری ها یک اسم دارند که یک رشته شامل حروف کوچک لاتین است. بجز *root* همه ی دایرکتوری ها یک دایرکتوری پدر دارند، که با .. نشان داده می شود.

دستور *cd* یک آرگومان می گیرد که یک مسیر در سیستم است، و دایرکتوری فعلی را به این دایرکتوری جدید تغییر می دهد. مسیر دایرکتوری دو مدل دارد، مدل *absolute* که از ریشه شروع می شود، و مدل *relative* که می تواند با نام دایرکتوری یا .. شروع شود، که .. دایرکتوری پدر دایرکتوری فعلی است.

دستور *pwd* هم آدرس *absolute* دایرکتوری فعلی را نشان می دهد و .. نباید داشته باشد. در ابتدای برنامه، آدرس فعلی ریشه است. همه دایرکتوری هایی که به صورت مستقیم یا غیر مستقیم در آرگومان دستور *cd* قرار می گیرند، در سیستم وجود دارند و تضمین می شود که در تست کیس ها دسترسی به پدر ریشه نخواهیم داشت.

ورودی

در خط اول یک عدد n ($1 \leq n \leq 50$) دریافت می کنید که n تعداد دستور ها است.

در n خط بعدی دستورات را دریافت می کنید که یا *cd* و یا *pwd* هستند.

آرگومان دستور *cd* حتماً طولی بزرگتر از صفر دارد و از کاراکتر های / و . و حروف کوچک لاتین تشکیل شده است، و . فقط برای نشان دادن پدر یک دایرکتوری استفاده می شود (یعنی نقطه تنها نداریم) و طول کل آدرس آرگومان حداکثر ۲۰۰ کاراکتر است و آدرس با / تمام نمی شود مگر آنکه کل آدرس فقط دایرکتوری *root* باشد.

خروجی

به ازای هر دستور *pwd* باید آدرس *absolute* دایرکتوری فعلی را چاپ کنید، که با /(اسلش) تمام می‌شود.

Input:

```
7
pwd
cd /home/farza
pwd
cd ..
pwd
cd farza/./parham
pwd
```

Output:

```
/
/home/farza/
/home/
/home/parham/
```

Input:

```
4
cd /a/b
pwd
cd ../a/b
pwd
```

Output:

```
/a/b/
/a/a/b/
```

خراب کاری

فرزاد اخیرا در دانشگاه، ساختمان داده *binary heap* را یاد گرفته، برای تمرین، برنامه‌ی یک *binray heap* را نوشته و این عملیات را برایش تعریف کرده است:

— عدد داده شده را در *heap* قرار بدهد

— مقدار مینیمم *heap* را بگیرد

— مقدار مینیمم *heap* را حذف کند

حالا فرزاد برای اینکه خوب این ساختار داده را یاد بگیرد، یک *heap* خالی را گرفته و روی آن عملیات بالا را انجام می‌دهد، علاوه بر آن، از همه این عملیات با دقت و به فرمت زیر *log* می‌گیرد:

— insert x

— getMin x

— removeMin

که این عملیات به ترتیب همان کار های لیست گفته شده در بالا را انجام می‌دهند. فرزاد دقت کرده که همه عملیاتی که انجام می‌دهد صحیح باشد، یعنی مثلا اگر می‌خواهد *removeMin* یا *getMin* را استفاده کند، مطمئن است که *heap* خالی نیست، بنابراین، کل *log* او صحیح است.

دوست فرزاد به او زنگ می‌زند و فرزاد از خانه بیرون می‌رود، در غیاب فرزاد، برادر کوچک‌ترش، به اتاقش می‌آید و با بعضی از برگه های *log* فرزاد، قایق کاغذی درست می‌کند و با آنها بازی می‌کند، کمی بعد متوجه می‌شود که چه اشتباهی کرده است، حالا *log* فرزاد ممکن است درست نباشد، یعنی اگر به ترتیب آنها را روی یک *heap* خالی اجرا کند، برنامه فرزاد *crash* کند، یا اینکه جواب *getMin* در *log*، با مقدار مینیمم موجود در *heap* در یک مرحله متفاوت باشد.

برای همین، تصمیم می‌گیرد که هر چه زودتر قبل از اینکه فرزاد به خانه برگردد، این مشکل را حل کند، اما او نمی‌تواند به تنهایی این مشکل را حل کند، بنابراین از شما کمک خواسته که این کار را برای او انجام دهید. او می‌خواهد برای آنکه سریع‌تر مشکل حل شود، حداقل دستور های لازم به *log* اضافه شود، به طوری که خروجی همه *getMin* ها برابر مقدار موجود در *log* باشد و وقتی به دستورات *getMin* یا *removeMin* رسیدیم، *heap* خالی نباشد.

* دقت کنید که عملیاتی که به *log* اضافه می‌کنید، به هر جای *log* (اول، وسط و آخر) می‌تواند اضافه شود.

* مقادیر *heap* اعداد صحیح هستند. و مقدار آنها از 10^9 کمتر است

ورودی

در خط اول ورودی یک عدد صحیح n ($1 \leq n \leq 100,000$) داده می‌شود که تعداد دستورات باقی مانده در \log ناقص فرزاد است.

در n خط بعدی، n دستور می‌آید که فرمت آنها دقیقاً مانند فرمت توضیح داده شده برای \log است.

خروجی

خط اول خروجی شامل یک عدد m است، که حداقل تعداد دستورات در \log بعد از اضافه کردن دستورات جدید است به طوری که \log جدید صحیح باشد.

در m خط بعدی، باید دستورات \log جدید چاپ شود.

تضمین می‌شود که هر مساله حتماً جوابی با کمتر از $1,000,000$ عملیات دارد.

دقت کنید که ورودی مساله، زیرمجموعه‌ای از خروجی خواهد شد، به مثال‌های زیر توجه کنید.

Input:

2

insert 3

getMin 4

Output:

4

insert 3

removeMin

insert 4

getMin 4

Input:

4
insert 1
insert 1
removeMin
getMin 2

Output:

6
insert 1
insert 1
removeMin
removeMin
insert 2
getMin 2

سیستم ثبت نام

فرزاد با انفورماتیک دانشگاه همکاری می‌کند، و انفورماتیک قصد دارد سیستم ایمیل خود را بروز رسانی کند، برای این کار، از فرزاد خواسته اند تا به شکل زیر، سیستم ثبت نام را بروز رسانی کند:

هر وقت که یک کاربر جدید می‌خواهد ایمیل جدید ثبت کند، به سیستم یک درخواست شامل نام خود می‌فرستد، اگر این نام در دیتابیس وجود نداشت، اضافه می‌شود و کاربر پاسخ *OK* می‌گیرد، در غیر این صورت، سیستم باید یک نام جدید با استفاده از نام کاربر برایش ایجاد کند و به کاربر پیشنهاد دهد، به این صورت که یک عدد، با شروع از ۱ به انتهای نام کاربر چسبانده می‌شود، به صورتی که این نام در دیتابیس وجود نداشته باشد. یعنی اگر نام کاربر *parham* بود و نام های *parham ۱* و *parham ۲* در دیتابیس وجود داشتند، برنامه باید نام *parham ۳* را به کاربر پیشنهاد دهد و سپس آن نام را به دیتابیس اضافه کند.

اما روش نگهداری دیتا در سیستم قدیمی متفاوت بود. به این صورت که اسم یکسان را می‌پذیرفت و برای هر فرد یک *id* منحصر به فرد در نظر می‌گرفت، اما حالا دیگر به *id* ها نیازی نیست، برای همین، لیست اسم های موجود در دیتابیس اولیه به شما داده می‌شود، و شما باید این اسامی را هم مطابق سیستم جدید در دیتابیس داشته باشید. فرزاد هم طبق معمول از شما کمک خواسته است که این کار را برایش انجام دهید.

* اگر بیش از ۹ اسم یکسان داشتیم هم اعداد دو رقمی (۱۰ و ۱۱ و ...) به انتهای اسم می‌چسبند.

ورودی

در خط اول دو عدد m, n ($1 \leq m, n \leq 10000$) دریافت می‌کنید که n تعداد درخواست ها به سیستم و m تعداد کاربران در دیتابیس قدیمی است.

در خط دوم، m رشته که با فاصله از هم جدا شده اند می‌آید که لیست کل نام های کاربران در دیتابیس قدیم است.

در n خط بعدی به ازای هر خط یک رشته می‌آید که شامل حروف کوچک لاتین با طول کمتر از ۳۲ است

دقت کنید که در دیتابیس قدیم، نام های تکراری می‌تواند وجود داشته باشد.

خروجی

اگر نام در دیتابیس وجود داشت، طبق قانون توضیح داده شده، برنامه شما باید نام جدید را بسازد و آن را چاپ کند، اگر نام در دیتابیس وجود نداشت، باید *OK* چاپ شود.

Input:

4 5

parham farshad farhad farzad fardad

farzad

acaba

farzad

acab

Output:

farzad1

OK

farzad2

OK

Input:

6 3

first first first

first

first

second

second

third

third

Output:

first3

first4

OK

second1

OK

third1