

کوچک سازی

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

با دریافت یک رشته، بزرگترین کاراکتر آن را حذف کنید.

ورودی

در خط اول، n یا طول رشته‌ی مورد نظر و در خط بعد خود رشته نوشته می‌شود.

$$2 \leq n \leq 200000$$

خروجی

بزرگترین کاراکتر رشته‌ی ورودی را حذف کرده و رشته‌ی باقی مانده را در یک خط از خروجی استاندارد چاپ کنید.

توجه: در صورت وجود چند گزینه‌ی حذف، کاراکتری را حذف کنید که به ابتدای رشته نزدیک‌تر است.

ورودی نمونه ۱

3

aaa

خروجی نمونه ۱

aa

ورودی نمونه ۲

6

adbcda

خروجی نمونه ۲

abcda

یکسان سازی

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

برنامه‌ای بنویسید که با دریافت دو رشته با طول برابر، امکان ساخت دو رشته‌ی یکسان را از آن دو با انجام تعداد دلخواه از عملیات زیر روی هر یک مشخص کند:

در هر عملیات می‌توان هر دو کارکتر مجاور دلخواه از یکی از رشته‌ها را انتخاب کرده و مقدار یکی از آن دو کاراکتر را برابر دیگری قرار داد. برای هر یک از دو رشته می‌توان هر تعداد دلخواه این عملیات را انجام داد.

برای مثال اگر رشته $acbc$ باشد، عملیات زیر قابل انجام هستند:

$$(S2 = S1)abc$$

$$(S1 = S2)ccbc$$

$$(S3 = S2 \text{ or } S3 = S4)accc$$

$$(S2 = S3)abbc$$

$$(S4 = S3)acbb$$

راهنمایی: به متن سوال و عملیات بالا دقت کنید. شاید نکته‌ی خیلی ساده‌ای از آن‌ها قابل دریافت باشد.

ورودی

در خط اول n نوشته می‌شود که تعداد جفت رشته‌های ورودی را نشان می‌دهد.

$$1 \leq n \leq 50$$

در n خط بعد، در هر خط دو رشته با طول برابر L از حروف کوچک انگلیسی نوشته می‌شود که با فاصله جدا شده‌اند.

$$1 \leq L \leq 15$$

خروجی

در n خط از خروجی استاندارد، در خط i ام در صورت امکان یکسان‌سازی دو رشته‌ی متناظر با ورودی i ام عبارت YES و در غیر این صورت عبارت NO را چاپ کنید.

ورودی نمونه ۱

```
2
ekwbrtq odizvex
rpi iwt
```

خروجی نمونه ۱

```
YES
YES
```

ورودی نمونه ۲

```
3
xabb aabx
technocup technocup
a z
```

خروجی نمونه ۲

YES

YES

NO

خلاصه

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

با دریافت مجموعه‌ای از کلمات، برای کلماتی که طول آن‌ها بیشتر از ده می‌باشد، حروف اول و آخر را نگه داشته و به جای حروف میانی تعداد آن‌ها را قرار دهید.

ورودی

در خط اول n یا تعداد کلمات و در n خط بعد خود کلمات با حداکثر طول l نوشته می‌شوند.

$$1 \leq n, l \leq 1000$$

خروجی

در n خط از خروجی استاندارد کلمات ورودی را با ترتیب اولیه و به صورت خواسته شده چاپ کنید.

ورودی نمونه ۱

```
3
more
localization
industrial
```

خروجی نمونه ۱

```
more
l10n
industrial
```

ورودی نمونه ۲

```
2
read
abbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbba
```

خروجی نمونه ۲

```
read
a30a
```

مرتب‌سازی اسم‌ها

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

با دریافت تعدادی اسم نامنظم، ابتدا آن‌ها را طوری تغییر دهید که حرف اول هر کلمه از هر اسم uppercase باشد. سپس بر اساس حروف الفبا اسم‌ها را مرتب کنید.

توجه: حتماً از الگوریتم مرتب‌سازی درجی استفاده کنید. اگر دو کلمه‌ی اول برابر بودند، کلمات بعد را مقایسه کنید.

ورودی

در خط اول n یا تعداد اسمی نوشته می‌شود. در n خط بعد، n اسم با حداکثر طول l نوشته می‌شود که کلمات هر کدام با یک فاصله از یکدیگر جدا می‌شوند.

$$1 \leq n \leq 100$$

$$1 \leq l \leq 50$$

خروجی

در n خط از خروجی استاندارد، اسم‌ها را به صورت خواسته شده چاپ کنید.

ورودی نمونه ۱

3

moHamMaDREza aMIni

RozHIn dADkhaH

ArMin KalHoR

خروجی نمونه ۱

Armin Kalhor
Mohammadreza Amini
Rozhin Dadkhah

ورودی نمونه ۲

2
jaKe sIlveR PaUl
SeLENa AdaMS

خروجی نمونه ۲

Jake Silver Paul
Selena Adams

اعداد مختلط

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

در این سوال از شما خواسته شده است که با کمک ساختارها برنامه‌ای برای کار با اعداد مختلط بنویسید. ساختار زیر را در برنامه‌ی خود تعریف کرده و سپس با دریافت عناصر دو عدد مختلط و یک عمل مورد نظر، دو ساختار **complex** برای آن‌ها ایجاد کرده و پس از محاسبه‌ی عمل مورد نظر روی آن دو، برای جواب نیز یک ساختار **complex** ایجاد کرده و آن را نمایش دهد.

```

1 struct complex
2 {
3     int re;
4     int img;
5 };

```

- عمل مورد نظر جمع، تفریق یا ضرب می‌باشد. (که فرض شده است شما با آن‌ها آشنا هستید!!!)
- اعداد مختلط ورودی با الگوی $re + i * im$ نوشته می‌شوند حتی اگر ضرایب re و img صفر یا یک باشند.
- فرمت جواب به صورت $re + imgi$ است. در چاپ آن دقت کنید که i بعد از ضریب img نوشته شده است.

ورودی

ورودی شامل سه خط است. در دو خط اول اعداد مختلط با الگوی مشخص شده و در خط سوم عمل مورد نظر به صورت $+$ ، $-$ یا $*$ نوشته می‌شوند.

$$-100 < re, img < 100$$

خروجی

در یک خط از خروجی استاندارد با فرمت ذکر شده حاصل محاسبات را چاپ کنید. به نحوه‌ی نمایش اعداد مختلط در مثال‌ها اکیدا توجه کنید.

مثال

ورودی نمونه ۱

```
44+i*62
-44-i*62
+
```

خروجی نمونه ۱

```
0
```

ورودی نمونه ۲

```
-66-i*46
-98-i*46
-
```

خروجی نمونه ۲

```
32
```

ورودی نمونه ۳

```
28+i*25
-28-i*95
```

$+$

خروجی نمونه ۳

 $-70i$

ورودی نمونه ۴

 $80-i*16$ $-66+i*90$ $*$

خروجی نمونه ۴

 $-3840+8256i$

دفتر تلفن

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

برای ذخیره‌سازی مجموعه‌ای از داده‌ها در کنار یکدیگر فقط از آرایه استفاده نمی‌شود بلکه روش‌های بسیار زیاد دیگری نیز برای این کار وجود دارد. یکی از این روش‌ها استفاده از *ساختمان داده‌ی لیست پیوندی* است که در آن به جای نگهداری تمام عناصر به صورت پشت سر هم مانند آرایه، با استفاده از ساختارها هر عنصر یک اشاره‌گر (پیوند) به عنصر بعد از خود را نگهداری کند. به عبارتی یکی از فلسفه‌های استفاده از لیست پیوندی نگهداری *زنجیره‌ی داده* به جای *بلوک داده* است. با درنظر گرفتن این نکات می‌توان فرض کرد که برای ذخیره‌سازی هزار داده‌ی یک بایتی در آرایه باید حداقل یک کیلوبایت حافظه‌ی آزاد پیوسته وجود داشته باشد در حالی که با استفاده از لیست پیوندی نیازی به پیوسته بودن آن فضا نیست. لیست پیوندی مزایا و معایب خاص خود را دارد از جمله آن که برای دسترسی به عنصر *i*ام آرایه کافیهست از ابتدای آن *i* خانه پرش انجام داد در حالی که در لیست پیوندی، هر عنصر فقط عنصر بعد از خود را می‌شناسد و باید از ابتدا روی تمامی عناصر حرکت کرد تا به عنصر *i*ام رسید. در عین حال مدیریت پویای مجموعه در لیست پیوندی ساده‌تر است و برای اضافه کردن یک عنصر جدید به مجموعه، کافیهست آن را به آخرین عنصر لیست وصل کرد یعنی عنصر آخر دیگر به هیچ جا اشاره نکند بلکه به عنصر جدید به عنوان عنصر بعد از خود اشاره کند.

توجه: توضیحات ارائه شده برای حل این سوال کافی هستند. با این حال شما در درس‌های آینده به صورت عمقی‌تر با لیست پیوندی آشنا خواهید شد. برخی از پیاده‌سازی‌های آن نیز به صورت صریح ذکر نشدند و به عهده‌ی شما هستند، مثلاً آن که چه روش‌هایی برای دسترسی به عنصر آخر وجود دارد. در صورتی که به مطالعه‌ی بیشتر درمورد این ساختمان داده علاقه دارید، پیشنهاد می‌شود نیم نگاهی به این لینک داشته باشید.

قرار است یک دفتر تلفن برای ذخیره‌سازی شماره‌ی مخاطبین بنویسید که مجموعه‌ی مخاطب‌ها در آن با استفاده از لیست پیوندی ذخیره‌سازی می‌شود. هر مخاطب نام (رشته)، نام خانوادگی (رشته) و شماره‌ی تلفن (عدد صحیح) دارد. کد زیر شکل کلی ساختار مورد استفاده در برنامه‌ی شما را نشان می‌دهد. همان‌طور

که مشاهده می‌شود اشاره‌گر **next** برای هر مخاطب، به مخاطب بعد از او اشاره می‌کند. همچنین متغیر جهانی **first** اولین عنصر لیست پیوندی مخاطبین را نشان می‌دهد.

```

1 | struct contact
2 | {
3 |     //Any Type of Data Set
4 |     contact* next;
5 | };
6 | contact** first; //Or contact* first

```

با فرض آن که حداکثر طول رشته‌ها 20 باشد، برنامه‌ای بنویسید که دستورات زیر را تا زمان وارد شدن دستور خروج دریافت و اجرا کند:

- دستور **Add** که به شکل زیر اطلاعات مخاطب جدید را دریافت و او را به لیست پیوندی مخاطبان اضافه می‌کند.

A [first_name] [last_name] [number]

- دستور **Edit** که نام و نام خانوادگی یک مخاطب را دریافت کرده و شماره‌ی او را برابر شماره‌ی دریافت شده قرار می‌دهد. اگر مخاطبی با نام و نام خانوادگی مربوطه وجود نداشت باید عبارت Contact doesn't exist چاپ شود. تضمین می‌شود که هیچ دو مخاطبی نام و نام خانوادگی یکسان ندارند.

E [first_name] [last_name] [number]

- دستور **Number** که نام و نام خانوادگی یک مخاطب را دریافت کرده و شماره‌ی او را در یک خط چاپ می‌کند. اگر مخاطبی با نام و نام خانوادگی مربوطه وجود نداشت باید عبارت Contact doesn't exist چاپ شود.

N [first_name] [last_name]

- دستور Show که به شکل زیر، در هر خط اطلاعات هر یک از مخاطبان دفتر تلفن را با ترتیب ورود چاپ می‌کند. اگر در دفتر تلفن هیچ مخاطبی وجود نداشت باید عبارت Contact list is empty چاپ شود.

S

[first_name] [last_name] [number]

- دستور Exit که اجرای برنامه را به پایان می‌رساند.

X

توجه: کد شما بررسی خواهد شد و در صورت عدم پیاده‌سازی ساختار لیست پیوندی نمره‌ای از این سوال دریافت نخواهید کرد.

مثال

ورودی نمونه

S

E john wick 858

A ali alavi 117

N ali alavi

E ali alavi 243

N taghi taghavi

E ali alavi 666

A taghi taghavi 257

A naghhi naghavi 572

A reza hossein 851

S

A mohammad alavi 212

A taghi taghi 64

S

X

خروجی نمونه

Contact list is empty
Contact doesn't exist
117
Contact doesn't exist
ali alavi 666
taghi taghavi 257
naghi naghavi 572
reza hossein 851
ali alavi 666
taghi taghavi 257
naghi naghavi 572
reza hossein 851
mohammad alavi 212
taghi taghi 64