

تخریب پایگاه داده

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: 100 مگابایت

--الیوت جوانی گوشه‌گیر است که دستی بر برنامه‌نویسی و امنیت شبکه دارد. او به عنوان مهندس شبکه در شرکتی به اسم آل‌سیف در نیویورک مشغول به کار می‌باشد؛ اما داستان او به اینجا ختم نمی‌شود. او تفکرات عجیبی در ذهن خود دارد اما از آن‌جایی که به کسی اعتماد ندارد، در ذهنش با شما حرف می‌زند. او می‌گوید گروهی از افراد قدرتمند وجود دارند که این افراد به طور مخفی دنیا را اداره می‌کنند؛ گروهی که هیچکس از آنها اطلاعی ندارد؛ گروهی که هرکاری بخواهند می‌کنند و نیاز به اجازه هیچکس ندارند! همچنین او فکر میکند آن‌ها او را دنبال می‌کنند و می‌خواهد قبل از اینکه دستشان به او برسد، خودش اولین ضربه را به آن‌ها بزند. او قصد دارد این‌کار را از شرکتی که فکر میکند پوششی برای این گروه است، آغاز کند. او می‌خواهد برنامه‌ای بنویسد که اطلاعات تعدادی از پایگاه‌های داده این شرکت(که E Corp نام دارد) را رمزگذاری کند تا آن اطلاعات غیر قابل استفاده شود و آسیبی جدی به آن‌ها وارد کند و خودی نشان بدهد. الیوت به علت تبحر زیاد در برنامه‌نویسی، میتواند برنامه تخریب یک پایگاه داده را در یک خط کد بنویسد! همچنین واضح است که تخریب صفر پایگاه داده نیاز به هیچ کدی ندارد.از آنجایی که الیوت فردی خسته است ، بیش از 19 پایگاه داده نیز نمی‌تواند تخریب کند. اما برای تعداد بیشتر از یک پایگاه داده، رابطه‌ای برای تعداد خط کد مورد نیاز برای نوشتن برنامه نیاز است که به صورت زیر نوشته می‌شود:-----

$$T(n) = T(n - 1) + T(n - 2) + n^2 \quad , \quad 2 \leq n \leq 4$$

$$T(n) = T(n - 1) + T(n - 2) + 2n^2 \quad , \quad 5 \leq n \leq 19$$

$$T(n) = che\ khabare \quad , \quad 20 \leq n$$

$$T(0) = 0 \quad , \quad T(1) = 1$$

ورودی

ورودی تنها شامل یک خط است که در آن یک عدد طبیعی n آمده است.(تعداد پایگاه)

$$0 \leq n \leq 50$$

خروجی

خروجی برنامه تنها شامل یک عدد است که تعداد خط کد مورد نیاز را می‌گوید.

مثال

ورودی نمونه ۱

1

خروجی نمونه ۱

1

ورودی نمونه ۲

20

خروجی نمونه ۲

che khabare

حرف زدن بسه

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

بعد از یک مکالمه طولانی و طاقت‌فرسا، شخصی که برای قانع کردن الیوت آمده بود، به این نتیجه می‌رسد که عقیده او تغییرپذیر نیست و نمی‌توان از او استفاده کرد. برای همین تصمیم می‌گیرد از راه دیگری وارد شود. او به الیوت می‌گوید که با استفاده از این کامپیوتر قدیمی که مستقیماً به پایگاه‌داده‌های شرکت متصل است، کلید مربوط به رمزگشایی اطلاعات را وارد کند. مشخصاً الیوت زیر بار نمی‌رود و می‌پرسد که چرا باید این‌کار را انجام دهد؟ آن شخص هم او را تهدید می‌کند که اگر این‌کار را انجام ندهد، حادثه ناگواری در آپارتمان‌ش و شرکت آل‌سیف (که الیوت در آن کار می‌کند) رخ خواهد داد. الیوت که می‌داند با چه گروه خطرناکی مواجه است، چاره‌ای جز انجام این‌کار ندارد. او پشت کامپیوتر می‌نشیند و شروع به ساخت کلید می‌کند.

کلید مورد نظر یک برنامه است که n بار عدد 7 را با طول و عرض های $2(n - i)$ که در آن i از 0 تا $n - 1$ است، با * رسم می‌کند. الیوت موظف است که برای این‌کار از تابع بازگشتی استفاده کند.

ورودی

ورودی عدد صحیح n :

$$0 \leq n \leq 100$$

خروجی

خروجی متشکل از 7 های ستاره ای زیر هم است که با طول و عرض $2n$ شروع شده و به مرور کوچک می‌شوند تا به 7 با طول و عرض 2 کارکتر برسیم. برای فهم بهتر سوال به نمونه های ورودی و خروجی توجه کنید.

مثال

ورودی نمونه ۱

3

خروجی نمونه ۱

```
\*****
      \*
    \*
  \*
 \*
\*
\*
\****
  \*
 \*
 \*
 \**
 \*
```

ورودی نمونه 2

5

خروجی نمونه 2

```
/ *****  
                                     / *  
  
                                   / *  
  
                                 / *  
  
                              / *  
  
                            / *  
  
                          / *  
  
                        / *  
  
                      / *  
  
                    / *  
  
                  / *  
  
                / *  
  
              / *  
  
            / *  
  
          / *  
  
        / ****
```

```
/ *  
  
      / *  
  
    / *  
  
  / *  
  
/ *
```

```
/ *  
  
      / *  
  
    / *  
  
  / *  
  
/ *
```

```
/ ****
```

```
/ *  
  
      / *  
  
    / *  
  
  / *  
  
/ *
```

```
/ **  
  
/ *
```

برج هانوی

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

سه میله‌ی - میله‌ی مبدأ (A) ، میله‌ی کمکی (B) و میله‌ی مقصد (C) - و تعدادی دیسک در میله‌ی مبدأ داریم. هدف انتقال تمام دیسک‌ها از این میله به میله‌ی مقصد با رعایت دو شرط زیر است:

- در هر زمان فقط یک دیسک را می‌توان جابجا نمود.
- نباید در هیچ زمانی دیسکی بر روی دیسک با اندازه‌ی کوچکتر قرار بگیرد.

به طور حتم می‌توان با روش آزمون و خطا به نتیجه‌ی مطلوب رسید. اما هدف ما ارائه‌ی الگوریتمی برای انتقال دیسک‌ها با کمترین جابجایی ممکن است.

به عنوان مثال، اگر $n=2$ باشد:

(1) دیسک 1 را به میله‌ی B منتقل می‌کنیم ($A \rightarrow B$):

(2) دیسک 2 را به میله‌ی C منتقل می‌کنیم ($A \rightarrow C$):

(3) دیسک 1 را به میله‌ی C منتقل می‌کنیم ($B \rightarrow C$):

توجه داشته باشید که بر اساس قانون اول، نمی‌توان به غیر از بالاترین دیسک هر میله، به دیسک دیگری از آن دسترسی پیدا کرد.

برای اینکه بتوان از روش بازگشتی برای حل یک مسئله استفاده نمود، مسئله باید قابلیت خرد شدن به زیرمسئله‌هایی از همان نوع مسئله‌ی اصلی و اندازه‌ی کوچکتر را داشته باشد. این ویژگی در مورد مسئله‌ی برج هانوی صدق می‌کند.

ایده‌ی اصلی از آنجا ناشی می‌شود که برای جابجا کردن بزرگترین دیسک از میله‌ی A به میله‌ی C، ابتدا باید تمامی دیسک‌های کوچکتر به میله‌ی B منتقل شوند. پس از تمام شدن این مرحله، دیسک بزرگ را از میله‌ی A به میله‌ی C منتقل کرده و مجدداً به کمک میله‌ی A تمامی دیسک‌های میله‌ی B را به میله‌ی C منتقل می‌کنیم. پس به طور خلاصه می‌توان گفت:

مرحله‌ی یک: $n-1$ دیسک بالایی میله‌ی مبدأ با شرایط ذکر شده و به کمک میله‌ی C به میله‌ی B منتقل می‌شوند.

مرحله‌ی دو: بزرگترین دیسک از میله‌ی مبدأ به میله‌ی مقصد منتقل می‌شود.

مرحله‌ی سه: $n-1$ دیسک میله‌ی B با کمک گرفتن از میله‌ی A به میله‌ی مقصد منتقل می‌شوند.

می‌بینیم که توانستیم عملیات جابجا کردن n دیسک را به دو عملیات مشابه ولی با اندازه‌ی کمتر و یک عملیات ساده تقسیم کنیم.

ورودی

ورودی شامل عدد صحیح n که تعداد دیسک را مشخص می‌کند.

$$1 \leq n \leq 20$$

خروجی

خروجی شامل خط‌هایی که به ترتیب نشان می‌دهد که از دیسکی از کدام میله برداشته و به کدام میله می‌رود.

مثال

ورودی نمونه ۱

خروجی نمونه ۱

A --> B
A --> C
B --> C

ورودی نمونه ۲

3

خروجی نمونه ۲

A --> C
A --> B
C --> B
A --> C
B --> A
B --> C
A --> C

سورت سریع

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

مرتب سازی سریع، یکی از الگوریتم‌های مرتب‌سازی است که به‌دلیل مصرف حافظه کم، سرعت اجرای مناسب و پیاده‌سازی ساده بسیار مورد قبول واقع شده‌است.

هر پیاده‌سازی این الگوریتم به‌صورت کلی از دو بخش تشکیل شده‌است. یک بخش تقسیم‌بندی آرایه (partition) و قسمت مرتب کردن. روش مرتب‌سازی سریع (Quick Sort) یکی از الگوریتم‌های مشهور مرتب‌سازی داده‌ها است. این الگوریتم طی مراحل بازگشتی زیر یک روش تقسیم و غلبه برای مرتب کردن داده‌ها ارائه می‌نماید:

۱. انتخاب عنصر محوری: یکی از عناصر آرایه به عنوان عنصر محوری (pivot) – به عنوان مثال عنصر اول – انتخاب می‌شود.
۲. تقسیم آرایه: چینش عناصر آرایه به قسمی تغییر داده می‌شود که تمامی عناصر کوچکتر یا مساوی محور در سمت چپ آن، و تمامی عناصر بزرگتر در سمت راست آن قرار بگیرند. این دو قسمت زیر آرایه‌های چپ و راست نامیده می‌شوند.
۳. مرتب‌سازی بازگشتی: زیرآرایه‌های چپ و راست به روش مرتب‌سازی سریع مرتب می‌شوند.

ورودی

در این قسمت ابتدا سائز آرایه ورودی دریافت می شود و سپس اجزای آرایه یکی یکی داده می شوند.

خروجی

در این قسمت ، آرایه سورت شده در یک خط نمایش داده می شود.

مثال

ورودی نمونه ۱

10
9
8
7
6
5
4
3
2
1
0

خروجی نمونه ۱

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

ورودی نمونه ۲

2
1
0

خروجی نمونه ۲

0 1

سایفر سزار

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

نقل شده است که ژولیوس سزار برای ارسال پیام های مهم نظامی اش هر حرف را به اندازه ۳ تا انتقال می‌داده! مثلا به جای A از حرف D و به جای حرف Z از حرف C استفاده می‌کرده. برنامه‌ای بنویسید که نام فایل و میزان دلخواه انتقال را از کاربر بگیرد، محتوی فایل را بخواند، آن را رمز کند و فایل رمز شده را ذخیره کند.

دقت کنید که این سوال توسط TA ها تصحیح می‌شود. شما باید کد زده شده را به همراه یک نمونه فایل ورودی و خروجی به صورت zip آپلود کنید.

ورودی

ورودی باید اسم فایل مورد نظر و میزان انتقال دلخواه باشد.

خروجی

خروجی برنامه‌ی شما باید شامل یک فایل باشد که عبارت رمز نگاری شده در خط اول آن و میزان انتقال در خط دوم آن نوشته شده است.

مثال

ورودی نمونه

```
myMessage.txt
3
```

فایل ورودی نمونه (myMessage.txt)

```
Dastoor midaham moteghaleban edam shavand
```

فایل خروجی نمونه (myMessageEnc.txt)

```
Gdvwrrru plgdkdp prwhjkdohedq hgdp vkdydqg
3
```

اسم فایل ها می‌تواند دلخواه باشد

ماشین متنی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

در این برنامه از شما خواسته شده است تا یک ماشین متنی کوچک با مجموعه‌ای از قابلیت های ساده را پیاده‌سازی نمایید. روال کار بدین صورت است که در آغاز کار، برنامه یک فایل حاوی رشته متنی مورد نظر (به طول حداکثر 1000) و دستوراتی که باید روی متن اعمال شود را می‌خواند. سپس همه دستورات را روی رشته متنی اعمال می‌کند و متن نهایی را در یک فایل ذخیره می‌کند.

دستورات به شرح زیر هستند:

دستور ورودی	توضیحات دستور
SHIFT-R N	تمام کاراکترهای عبارت را به صورت چرخشی N واحد به سمت راست منتقل می‌کند.
SHIFT-L N	تمام کاراکترهای عبارت را به صورت چرخشی N واحد به سمت چپ منتقل می‌کند.
EXTEND N	به انتهای رشته موجود N کاراکتر جدید اضافه می‌کند و به عنوان مقدار پیشفرض کاراکترها، ستاره (*) قرار می‌دهد.
SHRINK N	از انتهای رشته، N کاراکتر حذف می‌کند. در صورتی که طول رشته کمتر از N بود، رشته حاصل یک رشته خالی خواهد بود.
REVERSE	رشته را معکوس می‌کند.
PUT I C	حرف مکان I ام رشته را با حرف C جایگزین می‌کند. توجه داشته باشید که شماره مکان‌ها از یک آغاز می‌شود و I همواره کوچکتر مساوی طول رشته خواهد بود.
PRINT	رشته فعلی را در فایل چاپ می‌کند و به خط بعد می‌رود.
EXIT	اتمام ادیت کردن متن و خروج از برنامه.

دقت کنید که این سوال توسط TA ها تصحیح می‌شود. شما باید کد زده شده را به همراه یک نمونه فایل ورودی و خروجی به صورت zip آپلود کنید.

ورودی

ورودی باید اسم فایل مورد نظر باشد.

خروجی

خروجی برنامه باید یک فایل باشد که در آن متن ادیت شده قرار دارد. دقت کنید که در فایل خروجی به ازای هر بار اعمال دستور Print، یک خط نوشته داریم.

مثال 1

ورودی نمونه

File.txt
فایل ورودی نمونه (File.txt)
Test PRINT SHRINK 20 PRINT

```
EXTEND 2
PRINT
EXIT
```

فایل خروجی نمونه (Out.txt)

```
Test

**
```

مثال 2

ورودی نمونه

```
File.txt
```

فایل ورودی نمونه (File.txt)

```
initial string
PRINT
EXTEND 2
SHIFT-R 3
PRINT
PUT 3 0
REVERSE
SHRINK 2
PRINT
EXIT
```

فایل خروجی نمونه (Out.txt)

```
initial string
g**initial strin
nirts laitinio
```

اسم فایل ها می‌تواند دلخواه باشد