



درس برنامه سازی پیشرفته

تمرین اول

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۰ - ۹۹

استاد:

دکتر محمد امین فضلی

مبحث:

مباحث مقدماتی جاوا

مهلت ارسال:

۲۴ اسفند - ساعت ۲۳:۵۹:۵۹

مسئول تمرین ها:

پرهام صارمی

مسئول تمرین اول:

سایه جارالهی

طراحان تمرین اول:

مریم سادات رضوی، محمد امین آریان، امیر حسین هادیان، محمد مهدی برقی، پرهام چاوشیان،
علیرضا هنرور، امیر مهدی کوششی، محمد حسین حاجی سید سلیمان

ویراستار فنی:

فاطمه عسگری



به موارد زیر توجه کنید:

- * به ازای هر سوال در سامانه‌ی کوئرا، یک بخش جداگانه برای بارگذاری برنامه شما وجود دارد. برنامه خود با پسوند java. را در بخش مربوط به هر سوال بارگذاری کنید.
- * ورودی و خروجی شما باید عیناً شبیه به نمونه‌های ورودی و خروجی باشد؛ لذا عبارت‌هایی همچون "Enter your number" را قبل از گرفتن ورودی نباید چاپ کنید.
- * پس از ارسال فایل مربوط به هر سوال، سامانه‌ی کوئرا به صورت لحظه‌ای برنامه شما را داوری کرده و نمره آن سوال را به شما اعلام می‌کند که در صورت کم بودن نمره‌تان، می‌توانید آن را تصحیح کرده و دوباره ارسال کنید.
- * هم‌فکری و هم‌کاری در پاسخ به تمرینات اشکالی ندارد و حتی توصیه نیز می‌شود؛ ولی پاسخ‌های ارسالی شما باید حتماً توسط خود شما نوشته شده باشد. در صورت هم‌فکری در مورد یک سوال، نام فرد دیگر را به صورت کامنت در ابتدای کد هر سوال بنویسید.
- * شما می‌توانید تمامی سوالات و ابهامات خود را در سایت کوئرا در بخش مشخص شده برای هریک از سوالات بپرسید.
- * مهلت ارسال تمرین تا ساعت ۲۴:۰۰ روز ۲۴ اسفند ۱۳۹۹ است.
- * رعایت قواعد کدنویسی تمیز در این تمرین اجباری بوده و ۳۰ درصد از نمره کل تمرین مربوط به این بخش است. در صورتی که در سوالی نمره صفر گرفته باشید، نمره کدنویسی صحیح آن سوال را دریافت نمی‌کنید. این بخش در تحویل حضوری و مطابق با نکات ذکر شده در کدنامه شماره ۱ و ۲ بررسی می‌شود.



۱ امین و معضل نام گذاری

... توماس جوانی قدبلند و کمی لاغر که در این لحظه حتی اسمش را نیز به خاطر ندارد، خود را در میان زمینی سرسبز و نسبتاً بزرگ می‌یابد. او بعد از مدتی فریاد بی‌حاصل و فکر کردن به اینکه چه اتفاقی برایش افتاده و اینجا کجاست، تصمیم می‌گیرد برای وضعیتش کاری انجام دهد. به همین دلیل راهی زمینی که در میانش بیدار شده می‌شود.

ابعاد زمین ۱۰۰ در ۱۰۰ می‌باشد (از خانه ۱ تا ۱۰۰). حرکت از مرکز زمین (۵۰, ۵۰) شروع شده و n حرکت انجام می‌شود. حرکات به عنوان ورودی به برنامه داده می‌شوند. این حرکات ابتدا در راستای محور x و سپس در راستای محور y انجام می‌شوند. همچنین حرکات در صورتی اجرا می‌شوند که از روی مسیر حرکات قبلی عبور نکنند. به عبارتی، از هر خانه نقشه فقط یک بار می‌توان عبور کرد، چون توماس می‌خواهد هرچه سریع‌تر به نقاط جدید زمین برود بلکه بتواند نشانه‌ای پیدا کند. در صورتی که هر حرکتی شروط بالا را نداشته باشد، اجرا نشده و حرکت بعدی بررسی می‌شود. به توماس کمک کنید تا مختصات نهایی خود را پیدا کند.

ورودی

در خط اول عدد مثبت n داده می‌شود که تعداد حرکتهایی است که انجام می‌شود. در هریک از n خط بعدی، دو عدد داده می‌شود که عدد اول مشخص‌کننده طول حرکت در محور x و عدد دوم مشخص‌کننده طول حرکت در محور y است. اعداد منفی نشان‌دهنده حرکت در خلاف جهت محور مختصات هستند.

$$1 \leq n \leq 100$$

خروجی

خروجی برنامه شامل ۲ عدد با یک فاصله و در یک خط است که به ترتیب مختصات x و y توماس بعد از n حرکت را نشان می‌دهد.



ورودی نمونه

5
0 2
2 0
1 1
0 -2
-3 0

خروجی نمونه

50 53

ورودی نمونه

11
10 15
9 -4
-20 0
-45 9
3 0
0 9
12 -4
-10 -10
0 0
-1 1
8 14

خروجی نمونه

73 57



۲ سرخ

توماس پس از ساعت‌ها این طرف و آن طرف پرسه زدن و پیدا نکردن راه نجات، ناگهان در میان چمن‌ها نامه‌ای قدیمی پیدا می‌کند. در این نامه نوشته شده است که اگر در مسابقه‌ای شرکت کند و موفق شود، سرخی برای رها شدن از این وضعیت و راه خروجی پیدا خواهد کرد.

مسابقه در n روز برگزار می‌گردد و در هر روز برای هر شرکت‌کننده تعدادی کلمه در نظر گرفته شده است. سپس از آن‌ها خواسته می‌شود تا کلماتی که به آن‌ها در طول روز نمایش داده می‌شود را در صورت وجود از فهرست کلمات خود خط بزنند. پس از اتمام n روز، شرکت‌کنندگان باید اشتراک باقی‌مانده کلمات n روز خود را بگیرند و به ترتیب عکس حروف الفبا، آن‌ها را گزارش کنند. در نهایت، برنده از بین کسانی که هیچ خطایی مرتکب نشده و تمام مراحل را با دقت و به درستی انجام داده باشند، قرعه‌کشی می‌شود.

از آنجایی که توماس بچه زیرک و باهوشی است، تصمیم می‌گیرد با نوشتن برنامه‌ای، این کار را انجام دهد تا احتمال برنده شدن خود را در مسابقه بالا ببرد. شما باید در نوشتن این برنامه به او کمک کنید.

(توجه کنید که تکرار کلمات و بزرگ یا کوچک بودن حروف تأثیری ندارند.)

ورودی

ابتدا در خط اول عدد طبیعی n به شما داده می‌شود. در ادامه n رشته به ترتیب در n خط داده می‌آید که کلمات هر رشته با فاصله از یکدیگر جدا شده‌اند. در n خط بعدی n رشته دیگر مشابه رشته‌های قبلی داده می‌شود که کلمات حذفی n روز هستند.

$$1 \leq n \leq 1000000000$$

خروجی

شما باید پس از حذف کلمات هر رشته و محاسبه اشتراک کل، رشته اشتراک را به ترتیب عکس حروف الفبا چاپ کنید. در صورتی که اشتراکی وجود نداشت عبارت Noting in common را چاپ کنید.



ورودی نمونه

3
ali hasan hossein sajjad bagher sadegh
hossein kazem reza hasan
ali hossein reza taghi naghi sajjad hasan
sajjad bagher
kazem ali
taghi

خروجی نمونه

hossein hasan

ورودی نمونه

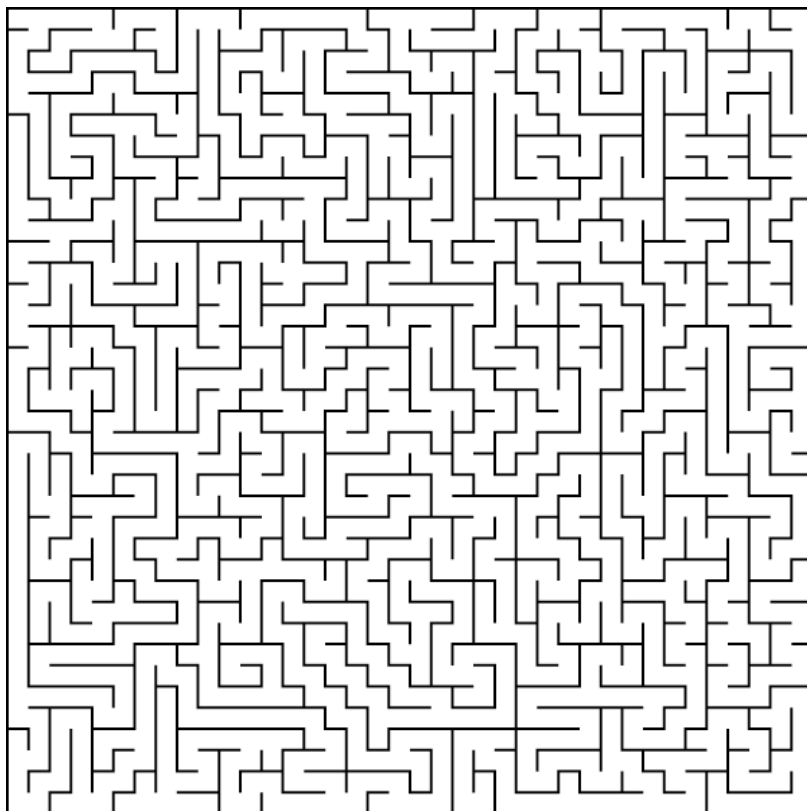
2
lion cow tiger bear
bear lion goat
lion
bear

خروجی نمونه

Nothing in common



۳ راه نجات



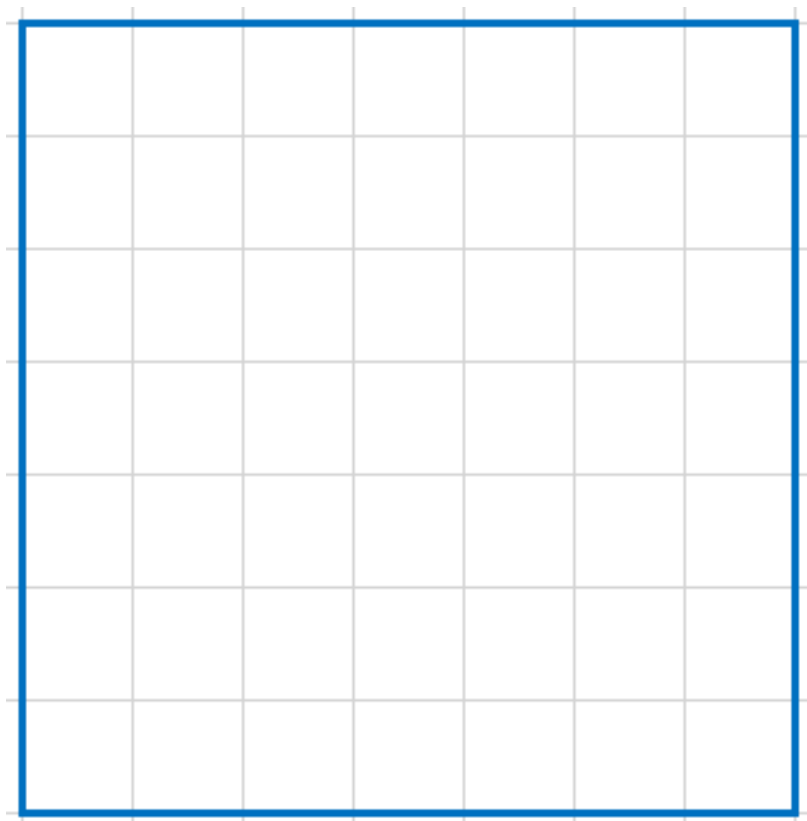
توماس بعد از تلاش های فراوان متوجه می شود که دقیقا در ابتدای یک ماز قرار دارد و تنها راه نجاتش گذر و عبور از این جدول ماز است. پس دست به کار می شود تا راه نجات را پیدا کند. او پس از چندین روز تلاش ناامید می شود اما درمی یابد که در مازی با طول m (تعداد ردیف های جدول ماز) و عرض n (تعداد ستون های جدول ماز) گیر افتاده است. در همین حین یادش می افتد که لپ تاپی را با خود به این سفر عجیب و غریب آورده است تا در مسافرت از دلایل هایش عقب نماند! پس به سرعت سراغ کیفش می رود و با هیجان درمی یابد که لپ تاپ هنوز سالم هست و کار می کند؛ اما لپ تاپ تنها ۱۵ درصد شارژ دارد. پس توماس از شما می خواهد هر چه سریع تر به او کمک کنید تا برنامه ای بنویسد که به او برای یافتن نقشه احتمالی این ماز کمک کند. توماس می داند در هر روز توانایی t بار بررسی کل ماز را دارد پس باید برنامه ای بنویسد که با گرفتن m (طول ماز) و n (عرض ماز)، t حالت و نقشه متفاوت از ماز را ترسیم کند تا او بتواند با آن نقشه ها به بررسی ماز بپردازد. نکته ای که باید به آن دقت کنید این است که توماس اصلا فرصت برای بررسی نقشه های



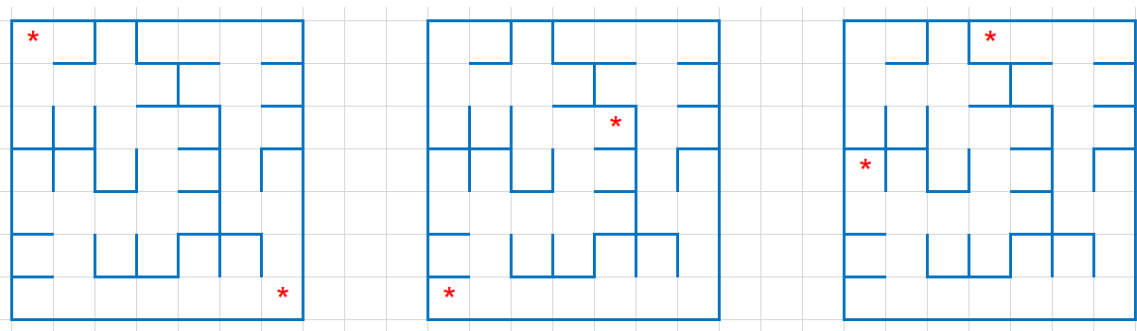
تکراری را ندارد و باید هر t نقشه خروجی برنامه شما حداقل در یک خانه تفاوت داشته باشند!

ماز باید ویژگی‌های زیر را داشته باشد:

- تمامی دیوارهای داخل ماز، حداقل به یک دیوار متصل هستند و دیواری وجود ندارد که به هیچ دیواری متصل نباشد. همچنین تمامی دیوارها به صورت مستقیم یا غیرمستقیم به دیواره اطراف ماز وصل هستند.
- بین هر دو خانه مربعی ماز، تنها یک مسیر یکتا وجود دارد.



دیواره دور ماز



بین هر دو خانه در ماز دقیقاً یک مسیر یکتا وجود دارد

این ماز قابل مدل کردن با گراف‌ها است. یکی از الگوریتم‌های پیمایش گراف که در حل این مسئله به شما کمک می‌کند، الگوریتم dfs است که برای فهم آن می‌توانید از این [لینک](#) و یا سایر منابع موجود در اینترنت استفاده کنید.

ورودی

ورودی به ترتیب شامل سه عدد m و n و t است. m عرض ماز و n طول آن است. همچنین t تعداد مازهای متفاوتی است که باید در خروجی چاپ شود.

$$1 \leq n, m, t \leq 100$$

خروجی

خروجی به تعداد t جدول به عرض $1+2*m$ و طول $1+2*n$ خواهد بود که هریک به فرمت زیر است. تمامی خانه‌های ماز (نقاط قرمز رنگ نشان داده شده) می‌بایست با $*$ در خروجی نشان داده شوند. تمام دیوارهای اطراف هر خانه از جدول ماز بدین صورت نشان داده می‌شوند که در صورت وجود دیوار آن را با 1 و در غیر این صورت (در صورتی که دیواری بین دو خانه وجود نداشت و امکان عبور وجود داشت) آن را با 0 نشان می‌دهیم. همچنین برای ماز یک ورودی و یک خروجی در نظر گرفته شده است که جای ثابتی دارند که در شکل نیز مشخص است. این ورودی و خروجی ماز با کاراکتر e نمایش داده می‌شوند. در تصویر زیر یک ماز $7*7$ و نحوه خروجی دادن آن مشخص شده است:



ورودی نمونه

3 6 3

خروجی نمونه

```
1e111111111111
1*0*0*0*0*0*1
1110111111101
1*1*0*1*0*1*1
1011111010101
1*0*0*0*1*0*1
11111111111e1
```

```
1e111111111111
1*1*0*0*0*0*1
1011111110101
1*0*0*0*0*1*1
1111111111101
1*0*0*0*0*0*1
11111111111e1
```

```
1e111111111111
1*0*0*1*0*1*1
1011101010101
1*0*1*0*1*0*1
1110111111101
1*0*1*0*0*0*1
11111111111e1
```

توجه داشته باشید که با توجه به اینکه مازها به طور رندوم ساخته می‌شوند، این تنها یک خروجی درست برای این ورودی است.



خروجی نمونه

[illegible]



۴ پیام ناشناس

پس از این که توماس موفق می‌شود به کمک شما به نقشه کلی ماز دست یابد، تصمیم می‌گیرد دل به دریا بزند و مسیر را آغاز کند. او اخیراً یک لینک پیام ناشناس برای شوخی با دوستانش برای خود درست کرده بود. غافل از اینکه همین لینک موجب دردسرش خواهد شد. در همان ابتدای مسیر توماس مرتباً پیام‌هایی از لینک ناشناس دریافت می‌کند که هر کدام ادعا می‌کنند نزدیک‌ترین خروجی ماز را یافته‌اند، اما توماس باهوش‌تر از این حرف‌هاست و درمی‌یابد که برخی از این پیام‌ها واقعی نیستند. او حسابی سردرگم شده است ولی خوش‌بختانه آشنایی مختصری با **API بات‌های تلگرام** (لینک صرفاً جهت اطلاعات بیشتر است.) دارد و قادر است اطلاعاتی از پیام‌ها استخراج کند. او می‌خواهد تا قبل از تاریک شدن هوا از طریق رشته‌ای که ربات ناشناس به او برگردانده، پیام‌های قابل اعتماد را شناسایی کند.

ورودی

ورودی شامل چهار خط است. خط اول یک رشته بلند است که حاوی تعدادی پیام معتبر و نامعتبر می‌باشد. خط دوم و سوم به ترتیب تاریخ شروع و پایان با فرمت YYYYMMDD هستند و خط چهارم شامل یک عدد اعشاری است که نشان‌دهنده موقعیت فعلی توماس است.

خروجی

خروجی پیام‌هایی هستند که ویژگی‌های زیر را داشته باشند:

- * روز ارسال آن‌ها در بازه بسته تاریخ شروع و پایان باشد.
- * موقعیت ارسال آن‌ها با موقعیت فعلی توماس حداکثر یک واحد تفاوت داشته باشد.
- * تمام فیلدهای آن‌ها در فرمت اعتبارسنجی مشخص شده صدق کند.



فرمت اعتبارسنجی

```
Message{ messageId=?, from=User{ firstName='?', isBot=
?, lastName='?', userName='?' }, date=?, text='?', locatio
n=? }
```

* هر پیام معتبر که فرستنده آن ربات باشد، خودش و پیام معتبر بعدی‌اش در رشته را از خروجی حذف می‌کند. (منظور از پیام معتبر صدق کردن در فرمت اعتبارسنجی است.)

نکات

۱. فیلدهای userName و lastName و text می‌توانند خالی باشند. (")
۲. فیلد isBot باید همواره مقداری برابر true یا false داشته باشد.
۳. تاریخ پیام باید با فرمت YYYYMMDDHHMMSS باشد. تضمین می‌شود تاریخ معتبر است.
۴. location که نشان‌دهنده موقعیت ارسال پیام است، باید عدد صحیح یا اعشاری باشد.
۵. در صورت وجود userName باید ۵ تا ۳۲ کاراکتر از حروف انگلیسی، اعداد و آندرلاین باشد، کاراکتر اول لزوماً باید حرف باشد و کاراکتر آخر نیز نمی‌تواند آندرلاین باشد.
۶. شناسه پیام معتبر باید دارای فرمت زیر و شامل سه بخش باشد:

%[1]-[2]\$[3]%

- در بخش [1] ارقام و اسپیس مجاز است.
- بخش [2] شامل ۵ حرف انگلیسی به جز حروف کلمه "thomas" (بزرگ و کوچک) می‌باشد و حرف اول بزرگ و بقیه حروف کوچک است.
- بخش [3] شامل ۲ یا ۴ رقم است.



مثال:

```
%45 225 3-Dklpe$65%
```

۷. تضمین می‌شود در متن فیلد هایی که داخل ” هستند، ’ وجود ندارد.

ورودی نمونه

```
helloMessage{ messageId=%98 73456 24 -Rlpnd$98%, from=User{ firstNam  
e='Mostafa', isBot=hehe, lastName='googooli', userName='' }, date=20200  
412063202, text='this is the body of message28!', location=136.43 }spam Me  
ssage{ messageId=%98 73456 24 -Rlpnd$98%, from=User{ firstName='Mort  
eza', isBot=true, lastName='golgoli', userName='' }, date=20200203124745,  
text='this is the body of message32!', location=136.43 }goodMessage{ mess  
ageId=% 5 -Rlpnd$0098%, from=User{ firstName='Teresa', isBot=false, las  
tName='magooli', userName='the_user54' }, date=20200203124745, text='t  
his is the body of message9!', location=136.43 }  
20200130  
20201201  
136
```

خروجی نمونه

توضیح

در این نمونه ۳ پیام در رشته مشاهده می‌شود. پیام اول به علت نامعتبر بودن فیلد - is Bot در فرمت اعتبارسنجی صدق نمی‌کند. پیام دوم و سوم در فرمت صدق می‌کنند ولی پیام دوم از طریق ربات ارسال شده و بنابراین خودش و پیام سوم را نیز از خروجی حذف می‌کند و در نتیجه هیچ پیامی در نهایت چاپ نمی‌شود.



ورودی نمونه

```
nice! Message{ messageId=%6-Rlpnd$9834%, from=User{ firstName='a', is
Bot=false, lastName='magooli', userName='the_user54' }, date=201804120
82501, text='this is the body of message!', location=136.43 }noo! Message{
messageId=%98 73456 24 -Rlpnd$98%, from=User{ firstName='Morteza', is
Bot=false, lastName='Akbari', userName='the_user54' }, date=2020020312
4745, text='this is the body of message26!', location=1.36.43 }ugly Message
{ messageId=%98 73456 24 -Rlpnd$918%, from=User{ firstName='Teresa',
isBot=false, lastName='magooli', userName='the_user54' }, date=20200412
063202, text='this is the body of message16!', location=136.43 }GOOD LUC
K!
20180312
20220101
137.0001
```

خروجی نمونه

```
*a magooli*
this is the body of message!
_08:25_
```

توضیح

در این نمونه نیز ۳ پیام مشاهده می‌شود. پیام اول کاملاً با فرمت مطابق است و شروط زمان و مکان نیز درباره آن صحیح هستند. پس در خروجی چاپ می‌شود. در پیام دوم فیلد location دارای عدد معتبری نیست. و در پیام سوم هم شناسه پیام مطابق فرمت خواسته شده نیست. (قسمت آخر آن ۳ رقم دارد)



۵ دیگه مرحله آخره

حال که توماس به مرحله آخر رسیده، یک کار خیلی مهم به او واگذار شده است تا انجام دهد. وظیفه توماس این است که روی یک سری اسناد (داک) که به مرور زمان به او داده می‌شود، تغییراتی را طبق دستورات انجام دهد و درست انجام دادن این دستورات باعث خواهد شد که او از مرگ رهایی پیدا کند و به زندگی جاودانه برسد!

دستورات

در ورودی سوال به شما دستوراتی داده می‌شود که باید روی داک‌ها اعمال کنید و خروجی مناسب را در صورت وجود نشان دهید. کلمات دستورها با یک اسپیس از هم جدا شده‌اند و هر یک در خط مجزایی داده می‌شوند. اسپیس‌های ابتدا و انتهای دستور، باعث اشتباه شدن دستور نمی‌شود و باید از آنها صرف نظر کنید. اگر دستوری که داده شده غلط بود و یا تعریف نشده بود، کافی است عبارت زیر را چاپ کنید.

invalid command!

در سراسر دستورات، در صورتی که فایلی با نام مشخص شده در دستور وجود نداشت **پیغام عدم وجود فایل** که به صورت زیر است چاپ می‌شود.

invalid file name!

ADD DOC [name]

این دستور به معنی اضافه کردن یک داک به داک‌های موجود است. بعد از آمدن این دستور بلافاصله در خط بعد متن داک در یک خط می‌آید. تضمین می‌شود که در نام داک اسپیس وجود ندارد. در صورتی که نام داک تکراری باشد، داک‌ی که از قبل موجود بوده پاک شده و داک جدید جایگزین آن می‌شود.



مثال:

```
ADD DOC 1.txt
salam @mid varam khob bashid!d
```

در این سوال تعدادی داک به شما داده می‌شود و با پردازش و تغییر روی آن‌ها، باید خروجی‌های مطلوب را به کاربر بدهید. هر داک یک اسم و یک محتوا دارد که اسم‌ها یکتا هستند اما متن داک‌ها می‌توانند تکراری باشند. متن داک‌ها در یک خط داده می‌شود. بین کلمات در داک یک یا چند اسپیس وجود دارد. پس از انجام پردازش‌ها بر روی داک، اسپیس‌ها حذف نمی‌شوند و به همان تعداد باقی می‌مانند. یک نمونه داک:

```
1.txt
salam doostan man AmirMahdi hastam
```

شما با گرفتن تعدادی داک و پردازش روی آن‌ها توسط دستوراتی که در ادامه آمده است، باید خروجی‌های مطلوب را چاپ کنید.

متن داک‌ها

همانطور که می‌دانید در صفحات داک علاوه بر متن، لینک‌ها و تصویرهایی نیز وجود دارند که به صورت متنی در داخل داک نوشته می‌شوند، ولی در صفحه به شکل همان لینک یا تصویر در می‌آیند. خود کلمات نیز ممکن است به شکل **bold** باشند یا بین کلمات نویز وجود داشته باشد.

برای تشریح هر کدام، متن‌های زیر را بخوانید.

- لینک: در میان کلمات داک ممکن است لینک‌هایی وجود داشته باشند. برای مثال:

[salam](#) ke shamel e ebarat e link shode va khode link ast

در واقع این لینک به صورت زیر در متن داک نوشته می‌شود:

```
[salam] (https://google.com) ke shamel e ebarat e link shode va khode
link ast
```



در واقع موقعی که داک را می‌خوانید باید به همان صورت

salam ke shamel e ebarat e link shode va khode link ast

ذخیره کنید. اما به فرمت بالا به شما داده خواهد شد. در این دستور تضمین می‌شود که لینک داده شده معتبر است و نیازی به چک کردن آن نیست. همچنین کلمه‌ای که به آن لینک می‌شود نویز و اسپیس ندارد.

- تصویر: در میان داک ممکن است تصاویری وجود داشته باشد. برای مثال:

in yek



ast

عکس ها نیز در واقع به صورت زیر در متن داک نوشته شده اند:

in yek [tasvir] (<https://quera.ir/qbox/view/TtRF7rH95i/pc.png>) ast

متن بالا را در واقع شما باید صورت

in yek tasvir ast

ذخیره کنید اما به صورت فرمت بالا به شما داده خواهد شد. در این دستور نیز تضمین می‌شود که لینک داده شده معتبر است و نیازی به چک کردن آن نیست. همچنین کلمه‌ای که به آن لینک می‌شود نویز و اسپیس ندارد.

- کلمات بولد



kalame **bold** shode

این کلمات به صورت زیر در داک موجود هستند:

```
kalame **bold** shode
```

متن بالا را در واقع شما باید صورت

kalame bold shode

ذخیره کنید اما به صورت فرمت بالا به شما داده خواهد شد.

- کلمات نویزدار: کلماتی که علامت هایی در بین شان دارند، کلمات نویزدار معنی می شوند.

```
ma@n hast—am ali bache ha
```

اگر کلماتی در بین شان نویز بیاید، شما باید از آن ها صرف نظر کنید. تضمین می شود که نویز در ابتدا و انتهای کلمات وجود ندارد.

به عنوان مثال در عبارت بالا، شما باید از کلمات ma@n و hast—am صرف نظر کنید زیرا کلمات نویز دار هستند. به عبارت دیگر جمله بالا بعد از اعمال نویز گیری به شکل زیر خواهد بود (به حذف نشدن اسپیس ها توجه کنید):

ali bache ha

مثال دیگر:

```
salA>m b^e ha@m#$gi haletoon khobe
```

ورودی داک به صورت بالا داده می شود اما باید به صورت زیر ذخیره شود:

haletoon khobe

خب حالا که فهمیدید این داک ها به چه شکل هستند، به ادامه دستورات می پردازیم.



RMV DOC [name]

این دستور به این معنی است که داک با نام مشخص شده باید برای همیشه پاک شود. اگر داک با این نام وجود نداشت پیغام عدم وجود فایل در یک خط مجزا چاپ می شود.

مثال:

RMV DOC 1.txt

RPLC [doc name] [word1,word2,...] [replace word]

این دستور به این معنی است که در داک با نام مشخص شده، باید آخرین بار حضور کلمات دسته اول را با کلمه دسته دوم جابجا کنید. در صورتی که داک با نام مشخص شده وجود نداشته باشد، پیغام عدم وجود فایل در یک خط مجزا چاپ می شود.

به مثال زیر توجه کنید:

RPLC s1.txt hello,no,bye salam

در صورتی که داک با نام s1.text محتوای زیر را داشته باشد:

hello hello to everyone bye bye nothing byee

پس از اجرای دستور، محتوای آن داک به صورت زیر می شود:

hello salam to everyone bye salam nothing byee

RPLC -ALL [word1,word2,...] [replace word]

این دستور به همان معنی دستور بالا است با این تفاوت که در تمام داک های موجود، این جایگزینی را انجام می دهد.



RMV WORD [doc name] [word]

این دستور در داک‌ی که نام آن ذکر شده است، تمام کلمات word را پاک می‌کند.

مثال: در صورتی که محتوای داک s1.txt به صورت زیر باشد:

salam saat 11 shabe salam khodafez salamm

با اجرای دستور زیر:

RMV WORD s1.txt salam

محتوای داک به صورت زیر می‌شود:

saat 11 shabe khodafez salamm

RMV WORD -ALL [word]

این دستور مانند دستور بالا است، با این تفاوت که در تمام داک‌های موجود، کلمه مورد نظر را پاک می‌کند.

ADD WORD [doc name] [word]

این دستور، کلمه ذکر شده را به انتهای داک با اسم مورد نظر اضافه می‌کند. در صورتی که داک‌ی با آن نام وجود نداشته، پیغام عدم وجود فایل چاپ می‌شود. تضمین می‌شود که کلماتی که برای اضافه شدن به داک می‌آیند، لینک و تصویر و بولد و نویزدار نیستند.

ADD WORD -ALL [word]

این دستور مانند دستور قبل عمل می‌کند و تنها تفاوت این است که در تمام داک‌های موجود، این تغییر را اعمال می‌کند.



```
FIND REP [doc name] [string]
```

این دستور، تعداد تکرارهای رشته ذکر شده را در متن داک به صورت زیر خروجی می دهد:

```
[string] is repeated [count] times in [doc name]
```

در صورتی که فایل با نام ذکر شده موجود نباشد، پیغام عدم وجود فایل چاپ می شود.

مثال:

```
FIND REP s1.txt AbcA
```

فرض کنید داک ۱ عبارتی مانند عبارت زیر داشته باشد. در صورتی که متن داک s1.txt به صورت زیر باشد:

```
AbcAbcAbcAbcA salam ok AbcA abc
```

خروجی به صورت زیر می شود:

```
AbcA is repeated 5 times in s1.txt
```

```
GCD [doc name]
```

این دستور، بزرگ ترین مقسوم علیه تمام اعداد موجود در متن را به انتهای متن اضافه می کند. در صورتی که فایل با نام ذکر شده موجود نباشد، پیغام عدم وجود فایل چاپ می شود.

مثال: پس از زدن دستور زیر:

```
GCD s1.txt
```

در صورتی که محتوای داک s1.txt به شکل زیر باشد



```
2 salam 3 bye 2 abcd 6
```

محتوای داک به شکل زیر تغییر می‌کند:

```
2 salam 3 bye 2 abcd 61
```

ب.م.م اعداد ۲ و ۳ و ۲ و ۱ و ۶ گرفته می‌شود و عدد ۱ به انتهای متن اضافه می‌شود.

```
FIND MIRROR [doc name] [char]
```

تعداد کلماتی را در متن پیدا میکند که کاراکتر قید شده در بین آنها است و دو عدد در دو سمت کاراکتر هستند که این دو عدد با یکدیگر برابرند.

مثال:

```
FIND MIRROR s1.txt b
```

به این معنی است که باید تعداد تمام کلماتی را پیدا کند که به صورت الگوی زیر هستند:

```
[adad]b[adad]
```

توجه کنید که دو عدد دو طرف کاراکتر یکسان هستند و اگر یکسان نباشد کلمه mirror به حساب نمی‌آید.

پس از پیدا کردن تعداد، آن را به صورت زیر در یک خط چاپ کند:

```
[count] mirror words!
```

```
FIND ALPHABET WORDS [doc name]
```

تعداد کلماتی را که در داک با نام ذکر شده فقط از حروف انگلیسی استفاده شده است را چاپ می‌کند.



مثال:

```
FIND ALPHABET WORDS 1.txt
```

فرمت چاپ کردن جواب به صورت زیر است:

```
[count] alphabetical words!
```

```
PRINT [doc name]
```

محتوای داک با نام ذکر شده را چاپ می‌کند.
توجه کنید که برای پرینت کردن محتوای داک شما باید داک را بدون لینک و تصویر و علامت بولد و نویز و همانطور که در بالا گفته شده است چاپ کنید.
در صورتی که داک با این نام وجود نداشت، خطای پیغام عدم وجود فایل چاپ می‌شود.

```
END
```

با آمدن این دستور برنامه پایان می‌یابد.

ورودی

دستورها هر یک در یک خط می‌آید. همچنین به گرفتن متن داک‌ها در دستور مربوطه توجه کنید.

خروجی

خواسته هر دستور (در صورت وجود) در یک خط چاپ می‌شود.



ورودی نمونه

```
ADD DOC 1.txt
I would like to know if there is **any** new information out there
about 2 th>subject or any new studies being conducted I am confident
in m>pediatrician and her communication with the people in Chapel Hill
bu>since this is a life-long
ADD DOC 2.txt
Hi I am doing a term paper on the syringe and I have found sominformation
It is my [link](https://google.com) that you can chek it out and this is so
**important**
PRINT 1.txt
PRINT 3.txt
PRINT 2.txt
RPLC 3.txt any,her ok
GCD 1.txt
PRINT 1.txt
FIND ALPHABET WORDS 2.txt
ADD DOC 3.txt
Recently I have come upon 1 body of literature which 3p3 coloclansing as a
vital's aid to 4p5 5 medicine through nutrition this 2 ![image1](https://goog
le/ok/bye) ![image2](https://google/ok/bye) could be helpful
PRINT 2.txt
FIND MIRROR 3.txt p
RMV WORD -ALL I
ADD WORD 3.txt I
PRINT 3.txt
GCD 3.txt
PRINT 3.txt
END
```

خروجی نمونه

I would like to know if there is any new information out there about 2 or any new studies being conducted I am confident in and her communication with the people in Chapel Hill this is a



invalid file name!

Hi I am doing a term paper on the syringe and I have found sominformation
It is my link that you can chek it out and this is so important

invalid file name!

I would like to know if there is any new information out there about 2 or any
new studies being conducted I am confident in and her communication with
the people in Chapel Hill this is a 2

30 alphabetical words!

Hi I am doing a term paper on the syringe and I have found sominformation
It is my link that you can chek it out and this is so important

1 mirror words!

Recently have come upon 1 body of literature which 3p3 colocleansing as a aid
to 4p5 5 medicine through nutrition this 2 image1 image2 could be helpfull
Recently have come upon 1 body of literature which 3p3 colocleansing as a aid
to 4p5 5 medicine through nutrition this 2 image1 image2 could be helpfulI1



ورودی نمونه

```
ADD DOC first.txt
I am 35 and my son is 5 we want to play a game that is on
[game](https://games.com) web wite Pox which I contracte>from my 5 year
old daughter this game is find overlap on OKOKOKOKO and print it
PRINT first.txt
FIND REP first.txt OKO
ADD DOC second.txt
this ![photo](https://heh.com) would be affected just amuchH Anthony Ch
ung-----+--C Case **Western** Reserve University School of Dentistry t
his is an email hac@po.CWRU and this is their [own](https://sharif.ir) webs
ite
PRINT second.txt
PRINT first.txt
FIND ALPHABET WORDS second.txt
RMV WORD second.txt would
ADD DOC third.txt
we 1 are 3 people and it is 11 pm and we are writing some test cases and it
will be good to know that writing test cases is so difficult for 3 people and we
must be 5 but we played a p2p and 2p2 game for Finding ErErErE from test
cases
RPLC third.txt test,cases ok
GCD third.txt
PRINT third.txt
END
```

خروجی نمونه

```
I am 35 and my son is 5 we want to play a game that is on game web wite Pox
which I my 5 year old daughter this game is find overlap on OKOKOKOKO
and print it
OKO is repeated 4 times in first.txt
this photo would be affected just amuchH Anthony Case Western Reserve
University School of Dentistry this is an email and this is their own website
I am 35 and my son is 5 we want to play a game that is on game web wite Pox
```



which I my 5 year old daughter this game is find overlap on OKOKOKOKO
and print it
25 alphabetical words!
we 1 are 3 people and it is 11 pm and we are writing some test cases and it
will be good to know that writing test cases is so difficult for 3 people and we
must be 5 but we played a p2p and 2p2 game for Finding ErErErE from ok
ok 1