

8월 18일 (수)

1. 백준 알고리즘 풀이 - 9093번 단어 뒤집기

문제

문장이 주어졌을 때, 단어를 모두 뒤집어서 출력하는 프로그램을 작성하시오. 단, 단어의 순서는 바꿀 수 없다. 단어는 영어 알파벳으로만 이루어져 있다.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다. 각 테스트 케이스는 한 줄로 이루어져 있으며, 문장이 하나 주어진다. 단어의 길이는 최대 20, 문장의 길이는 최대 1000이다. 단어와 단어 사이에는 공백이 하나 있다.

출력

각 테스트 케이스에 대해서, 입력으로 주어진 문장의 단어를 모두 뒤집어 출력한다.

예제 입력 1 복사

```
2
I am happy today
We want to win the first prize
```

예제 출력 1 복사

```
I ma yppah yadot
eW tnaw ot niw eht tsrif ezirp
```

파이썬은 다른 언어와 다르게 stack이 따로 구현 되어있지 않다. 대신 자료형 list 가 stack 형태로 만들어져 list를 이용해서 stack 을 사용한다.

2. Codecademy - Command Line, Git, Github - Learn Markdown

Article 1. What is Markdown?

- Intro
- Different between HTML and Markdown
- Benefits of using Markdown
- When to use Markdown
- Markdown Document Extension
- Markdown Applications
- The Markdown Parser
- The Markdown Flavors
- Conclusion

- **Intro**

마크다운을 이용하여 이목을 끄는 텍스트문서를 특정 에디터 없이 작성할 수 있다. 이는 html 로도 변환이 가능하다.

- **Different between HTML and Markdown**

마크다운은 html 의 대체품이 될 수 없다. 하지만 마크다운은 html 의 문법을 사용하며, 링크, 이미지 등을 첨부할 수 있다. html로 마크다운을 읽기 위해서는 Markdown Parser 가 필요하다.

- **Benefits of using Markdown**

html 대신 markdown 을 쓰는 이유는 문법이 간단해서 더 빠르다. html 보다 더 간편하게 쓸 수 있다.

```
My Todo List
```

```
=====
```

```
At the end of this week, I plan to:
```

1. Learn Markdown
2. Write Markdown
3. Share a Markdown note

```
My favorite quote is:
```

```
> If you didn't get it the first time  
> Do not despair  
> Try and try again  
> ~ Anonymous
```

The HTML equivalent to the above would be:

```
<h1>My Todo List</h1>  
<p>At the end of this <strong>week</strong>, I plan to:</p>  
<ol>  
  <li>Learn Markdown</li>  
  <li>Write Markdown</li>  
  <li>Share a Markdown note</li>  
</ol>  
<p>My favorite quote is:</p>  
<blockquote>  
  <p>  
    If you didn't get it the first time<br>  
    Do not despair<br>  
    Try and try again<br>  
    ~ Anonymous  
  </p>  
</blockquote>
```

My Todo List

At the end of this **week**, I plan to:

1. Learn Markdown
2. Write Markdown
3. Share a Markdown note

My favorite quote is:

If you didn't get it the first time

Do not despair

Try and try again

~ Anonymous

- **Markdown Document Extension**

일반적인 믿음과 달리, 마크다운은 문서형식이 아니다. 그러기에, .md 와 같은 엄격한 파일 확장 네이밍을 필요로 하지 않는다. 마크다운은 파일의 형식으로 넘겨지지 않지만, 깃허브와 같은 곳에는 마크다운을 .md 나 .markdown 으로 표시되는 것을 선호한다.

- The Markdown Parser

본래의 마크다운 파서는 Perl 로 쓰여졌다. 하지만 현재는 거의 대부분의 프로그램에 모두 구현되어 있다. 마크다운 파서는 웹에서도 자유롭게 이용이 가능하다. [StackEdit.io](#), [Dillinger](#), [Parse](#) and [Markdown to HTML Converter](#). 마크다운을 raw HTML 로도 바꿀 수 있다.

- Markdown Tutorial

```
# This is a H1 heading
## This is a H2 heading
### This is a H3 heading
#### This is a H4 heading
##### This is a H5 heading
##### This is a H6 heading
```

will render this:

This is a H1 heading

This is a H2 heading

This is a H3 heading

This is a H4 heading

This is a H5 heading

This is a H6 heading

For example:

```
* Markdown
+ HTML
- XML
```

produces this:

- Markdown
- HTML
- XML

Article 2. Mastering Markdown

- What is Markdown ?
- Examples
- Basic syntax
- GFM

- **What is Markdown ?**

마크다운은 웹에서 텍스트에 대한 스타일을 입힐 수 있는 방법이다. 깃허브의 대부분 장소에서 마크다운이 쓰인다.

⇒ Gists, 이슈나 Pull Requests 에서의 코멘트들, .md 파일

- Examples

⇒ Text

Text Lists Images Headers & Quotes Code Extras

It's very easy to make some words ****bold**** and other words **italic** with Markdown. You can even [link to Google!](http://google.com)

It's very easy to make some words **bold** and other words *italic* with Markdown. You can even [link to Google!](http://google.com)

⇒ Lists

Sometimes you want numbered lists:

1. One
2. Two
3. Three

Sometimes you want bullet points:

- * Start a line with a star
- * Profit!

Alternatively,

- Dashes work just as well
- And if you have sub points, put two spaces before the dash or star:
 - Like this
 - And this

Sometimes you want numbered lists:

1. One
2. Two
3. Three

Sometimes you want bullet points:

- Start a line with a star
- Profit!

Alternatively,

- Dashes work just as well
- And if you have sub points, put two spaces before the dash or star:
 - Like this

⇒ Images

If you want to embed images, this is how you do it:

```
![Image of Yaktocat]  
(https://octodex.github.com/images/yaktocat.png)
```

If you want to embed images, this is how you do it:



⇒ Header & Quotes

Structured documents

Sometimes it's useful to have different levels of headings to structure your documents. Start lines with a `#` to create headings. Multiple `##` in a row denote smaller heading sizes.

This is a third-tier heading

You can use one `#` all the way up to `#####` six for different heading sizes.

If you'd like to quote someone, use the > character before the line:

> Coffee. The finest organic suspension ever devised... I beat the Borg with it.

> - Captain Janeway

Structured documents

Sometimes it's useful to have different levels of headings to structure your documents. Start lines with a `#` to create headings. Multiple `##` in a row denote smaller heading sizes.

This is a third-tier heading

You can use one `#` all the way up to `#####` six for different heading sizes.

If you'd like to quote someone, use the > character before the line:

Coffee. The finest organic suspension ever devised... I beat the Borg with it. - Captain Janeway

⇒ Code

Text Lists Images Headers & Quotes **Code** Extras

There are many different ways to style code with GitHub's markdown. If you have inline code blocks, wrap them in backticks: ``var example = true``. If you've got a longer block of code, you can indent with four spaces:

```
    if (isAwesome){
      return true
    }
```

GitHub also supports something called code fencing, which allows for multiple lines without indentation:

```
```
if (isAwesome){
 return true
}
```
```

And if you'd like to use syntax highlighting, include the language:

```
```javascript
if (isAwesome){
 return true
}
```
```

There are many different ways to style code with GitHub's markdown. If you have inline code blocks, wrap them in backticks:

`var example = true`. If you've got a longer block of code, you can indent with four spaces:

```
if (isAwesome){  
  return true  
}
```

GitHub also supports something called code fencing, which allows for multiple lines without indentation:

```
if (isAwesome){  
  return true  
}
```

And if you'd like to use syntax highlighting, include the language:

```
if (isAwesome){  
  return true  
}
```

⇒ Extras

GitHub supports many extras in Markdown that help you reference and link to people. If you ever want to direct a comment at someone, you can prefix their name with an @ symbol: Hey @kneath — love your sweater!

But I have to admit, tasks lists are my favorite:

- [x] This is a complete item
- [] This is an incomplete item

When you include a task list in the first comment of an Issue, you will see a helpful progress bar in your list of issues. It works in Pull Requests, too!

And, of course emoji!

GitHub supports many extras in Markdown that help you reference and link to people. If you ever want to direct a comment at someone, you can prefix their name with an @ symbol: Hey @kneath — love your sweater!

But I have to admit, tasks lists are my favorite:

- ☒ This is a complete item
- ☐ This is an incomplete item

When you include a task list in the first comment of an Issue, you will see a helpful progress bar in your list of issues. It works in Pull Requests, too!

And, of course emoji!

- Basic syntax

Headers

```
# This is an <h1> tag
## This is an <h2> tag
##### This is an <h6> tag
```

Emphasis

```
*This text will be italic*
_This will also be italic_

**This text will be bold**
__This will also be bold__

_You **can** combine them_
```

Lists

Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered

```
1. Item 1
1. Item 2
1. Item 3
  1. Item 3a
  1. Item 3b
```


Images

```
![GitHub Logo](/images/logo.png)
Format: ![Alt Text](url)
```

Links

```
http://github.com - automatic!
[GitHub](http://github.com)
```

Blockquotes

```
As Kanye West said:

> We're living the future so
> the present is our past.
```

Inline code

```
I think you should use an
`<addr>` element here instead.
```

- **GFM (Github Flavored Markdown 깃허브 풍미를 들인 마크다운)**

깃허브는 추가적인 유용한 특징을 가진 세트를 제공하는 깃허브 고유의 마크다운 신택스 버전을 사용한다. 이는 깃허브에서의 작업을 더 쉽게 해준다.

GMF의 몇몇 기능들은 descriptions와 issues, Pull Requests의 Comment에서만 사용이 가능하다.

@mention뿐만 아니라 SHA-1 해쉬, 이슈, pull requests들을 포함한다. Task Lists는 또한 Gist comments나 Gist Markdown files에서 이용이 가능하다.

⇒ Syntax highlighting

Syntax highlighting

Here's an example of how you can use syntax highlighting with [GitHub Flavored Markdown](#):

```
```javascript
function fancyAlert(arg) {
 if(arg) {
 $.facebox({div:'#foo'})
 }
}
```
```

You can also simply indent your code by four spaces:

```
function fancyAlert(arg) {
    if(arg) {
        $.facebox({div:'#foo'})
    }
}
```

Here's an example of Python code without syntax highlighting:

```
def foo():
    if not bar:
        return True
```

⇒ Task Lists

Task Lists

- [x] @mentions, #refs, [links](), **formatting**, and tags supported
- [x] list syntax required (any unordered or ordered list supported)
- [x] this is a complete item
- [] this is an incomplete item

If you include a task list in the first comment of an Issue, you will get a handy progress indicator in your issue list. It also works in Pull Requests!

⇒ Tables

Tables

You can create tables by assembling a list of words and dividing them with hyphens `-` (for the first row), and then separating each column with a pipe `|`:

```
First Header	Second Header
Content from cell 1 | Content from cell 2
Content in the first column | Content in the second column
```

Would become:

| First Header | Second Header |
|-----------------------------|------------------------------|
| Content from cell 1 | Content from cell 2 |
| Content in the first column | Content in the second column |

⇒ SHA references

SHA references

Any reference to a commit's [SHA-1 hash](#) will be automatically converted into a link to that commit on GitHub.

```
16c999e8c71134401a78d4d46435517b2271d6ac
mojombo@16c999e8c71134401a78d4d46435517b2271d6ac
mojombo/github-flavored-markdown@16c999e8c71134401a78d4d46435517b2271d6ac
```

⇒ Issue references within a repository

Issue references within a repository

Any number that refers to an Issue or Pull Request will be automatically converted into a link.

```
#1  
mojombo#1  
mojombo/github-flavored-markdown#1
```

⇒ Username mentions

Username @mentions

Typing an `@` symbol, followed by a username, will notify that person to come and view the comment. This is called an “@mention”, because you’re *mentioning* the individual. You can also @mention teams within an organization.

⇒ Automatic linking for URLs

Automatic linking for URLs

Any URL (like `http://www.github.com/`) will be automatically converted into a clickable link.

⇒ Strikethrough

Strikethrough

Any word wrapped with two tildes (like `~~this~~`) will appear crossed out.

— --

⇒Emoji

Emoji

GitHub supports [emoji](#)!

To see a list of every image we support, check out the [Emoji Cheat Sheet](#).

<https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md> emoji 치트시트

Article 3. Markdown Cheatsheet

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Article 4. Markdown and Visual Studio Code

<https://code.visualstudio.com/docs/languages/markdown>

Article 5. How to write a good README for your GitHub project ?

리드미 잘 작성하는 방법

<https://bulldogjob.com/news/449-how-to-write-a-good-readme-for-your-github-project>

오늘의 단어

- lenient : 관대한