Implement your own strtok function

Description

Write a function char *my_strtok(char *str, char *delim) to extract tokens from strings.

On the first call to my_strtok(), the string str will be parsed and return the first token, separated by any char in the delim (to delimit the tokens a nd be ignored).

Each subsequent calls should parse the same string until all tokens are processed, where a NULL is returned. (Hint: use a static string pointer i n your function.)

Example:

str = “Happy Birthday!”

1st call my_strtok(str, “ .!”) return “Happy”

2nd call my_strtok(NULL, “ .!”) return “Birthday”

3rd call my_strtok(NULL, “ .!”) return NULL

`Note: You MUST implement your own function or you get 0 for calling strtok, strtok_s, and strtok_r in your code.`

Input

An integer t representing the number of test cases, followed by 2 x t lines.

For every 2 lines, first line contains char *str, second line contains char *delim

Both strings are enclosed in double quotes. Ex. "BanG Dream! It's MyGO!!!!!"

Constraints:

- 1 <= t <= 100
- str <= 100(Including double quotes)
- delim <= 10(Including double quotes)

Output

For every test case first print "Test case" and its number, then a newline in the end. Ex. "Test case 1\n", "Test case 2\n", etc. The number starts from 1 and increases in ascending order.

Next print out all the tokens, where every token is separated by a newline.

You don't need to print anything if token is NULL.

Sample Input 1

```
3
"Happy Birthday!"
" .!"
"BanG Dream! It's MyGO!!!!!"
"(^_-)!"
"Goblin. Together. Strong."
"!@#$^&*"
```

Sample Output 1

```
Test case 1
Happy
Birthday
Test case 2
BanG Dream
 It's MyGO
Test case 3
Goblin. Together. Strong.
```

Hint

use a static string pointer in your function.

Information

ID	final_3
Time Limit	1000MS
Memory Limit	256MB
IO Mode	Standard IO
Created By	stellaglow1122
Level	Low
Score	100
Tags	Show

Statistic Details