Validating and Parsing Email Addresses ★

84/115 challenges solved

Rank: 17722 | Points: 1245

Your Validating and Parsing Email Addresses submission got 20.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

A valid email address meets the following criteria:

- It's composed of a username, domain name, and extension assembled in this format: `username@domain.extension`
- The username starts with an English alphabetical character, and any subsequent characters consist of one or more of the following: alphanumeric characters, `-`,`.`, and `_`.
- The domain and extension contain only English alphabetical characters.
- The extension is 1, 2, or 3 characters in length.

Given n pairs of names and email addresses as input, print each name and email address pair having a valid email address on a new line.

Hint: Try using Email.utils() to complete this challenge. For example, this code:

```
import email.utils
print email.utils.parseaddr('DOSHI <DOSHI@hackerrank.com>')
print email.utils.formataddr(('DOSHI', 'DOSHI@hackerrank.com'))
```

produces this output:

```
('DOSHI', 'DOSHI@hackerrank.com')
DOSHI <DOSHI@hackerrank.com>
```

Input Format

The first line contains a single integer, n, denoting the number of email address.

Each line i of the n subsequent lines contains a name and an email address as two space-separated values following this format:

```
name <user@email.com>
```

Constraints

- $0 < n < 100$

Output Format

Print the space-separated name and email address pairs containing valid email addresses only. Each pair must be printed on a new line in the following format:

```
name <user@email.com>
```

You must print each valid email address in the same order as it was received as input.

Sample Input

```
2
DEXTER <dexter@hotmail.com>
VIRUS <virus!@variable.:p>
```

Sample Output

```
DEXTER <dexter@hotmail.com>
```

Explanation

dexter@hotmail.com is a valid email address, so we print the name and email address pair received as input on a new line.

virus!@variable.:p is not a valid email address because the username contains an exclamation point (`!`) and the extension contains a colon (`:`). As this email is not valid, we print nothing.

Change Theme Language Python 3

```python
import email.utils
import re

def is_valid_email(text):
    pattern = r'^[a-z][a-z0-9\-_\.]+[@][a-z]+[.][a-z]{1,3}$'

    return re.search(pattern, text)

if __name__ == '__main__':
    contacts = [input() for _ in range(int(input()))]

    for c in contacts:
        result = email.utils.parseaddr(c)
        if is_valid_email(result[1]):
            result = email.utils.formataddr(result)
            print(result)

```

Line: 17 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!
84/115 challenges solved.

73%

Python ★★★★★

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Earn a certificate in Python

Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

Get Certified

✓ Test case 0

Compiler Message

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

✓ Test case 6

```
Success
```

Input (stdin) Download

```
2
DEXTER <dexter@hotmail.com>
VIRUS <virus!@variable.:p>
```

Expected Output Download

```
DEXTER <dexter@hotmail.com>
```