

Validating and Parsing Email Addresses *



Your Validating and Parsing Email Addresses submission got 20.00 points.

X

Problem Submissions Leaderboard Editorial A

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 _.

Try the next challenge | Try a Random Challenge

- 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

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

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
 1
 2
      import re
      # Define regex pattern for a valid email address
      \label{eq:compile} \\ \texttt{email\_pattern} = \texttt{re.compile}(\texttt{r'}<[\texttt{a-zA-Z}](\texttt{w.-}]+@[\texttt{a-zA-Z}]+\\ \\ \\ \texttt{(a-zA-Z)}\{1,3\}>') \\ \\ \end{aligned}
 6
 7
      # Read input
 8
      n = int(input())
 9
      for _ in range(n):
10
11
           name, email = input().split()
12
            if email_pattern.fullmatch(email):
13
                 print(name, email)
14
```

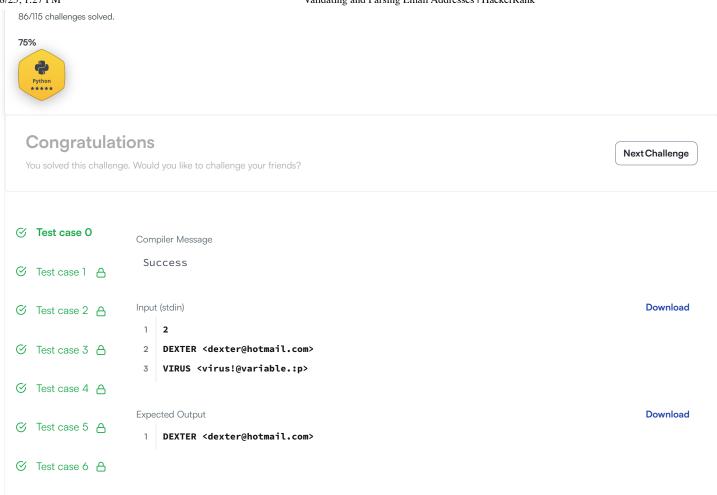
EMACS

Line: 14 Col: 27

Pupload Code as File Test against custom input

Run Code Submit Code

You have earned 20.00 points!



Blog | Scoring | Environment | FAQ | About Us | Helpdesk | Careers | Terms Of Service | Privacy Policy