Validating Email Addresses With a Filter

HackerRank

You are given an integer N followed by N email addresses. Your task is to print a list containing only *valid* email addresses in lexicographical order.

Valid email addresses must follow these rules:

- It must have the *username@websitename.extension* format type.
- The username can only contain letters, digits, dashes and underscores $[a-z], [A-Z], [0-9], [\_-]$.
- The website name can only have letters and digits $[a-z], [A-Z], [0-9]$.
- The extension can only contain letters $[a-z], [A-Z]$.
- The maximum length of the extension is 3.

Concept

A *filter* takes a function returning *True* or *False* and applies it to a sequence, returning a list of only those members of the sequence where the function returned *True*. A *Lambda* function can be used with filters.

Let's say you have to make a list of the squares of integers from 0 to 9 (both included).

```
>> l = list(range(10))
>> l = list(map(lambda x:x*x, l))
```

Now, you only require those elements that are greater than 10 but less than 80.

```
>> l = list(filter(lambda x: x > 10 and x < 80, l))
```

Easy, isn't it?

Example

Complete the function *fun* in the editor below.

fun has the following paramters:

- *string s:* the string to test

Returns

- *boolean:* whether the string is a valid email or not

Input Format

The first line of input is the integer N, the number of email addresses.
N lines follow, each containing a string.

Constraints

Each line is a non-empty string.

Sample Input

```
3
lara@hackerrank.com
brian-23@hackerrank.com
britts_54@hackerrank.com
```

Sample Output

```
['brian-23@hackerrank.com', 'britts_54@hackerrank.com', 'lara@hackerrank.com']
```