[CS 11] Prac 10g – Tabulation

Problem Statement

As part of a terminal-based app you are developing, you are given some numerical data arranged in a table, and you would like to display them nicely in the terminal. For example, a possible display could be:

```
|  n | fib(n) |  factorial(n) |
|  0 |      0 |             1 |
|  1 |      1 |             1 |
|  2 |      1 |             2 |
|  3 |      2 |             6 |
|  4 |      3 |            24 |
|  5 |      5 |           120 |
|  6 |      8 |           720 |
|  7 |     13 |          5040 |
|  8 |     21 |         40320 |
|  9 |     34 |        362880 |
| 10 |     55 |       3628800 |
| 11 |     89 |      39916800 |
| 12 |    144 |     479001600 |
| 13 |    233 |    6227020800 |
| 14 |    377 |   87178291200 |
| 15 |    610 | 1307674368000 |
```

Note that the separators and borders can be customized. For example, here's the same data with separator `::` and borders `$::` and `::$`:

```
$::   n :: fib(n) ::  factorial(n) ::$
$::   0 ::      0 ::             1 ::$
$::   1 ::      1 ::             1 ::$
$::   2 ::      1 ::             2 ::$
$::   3 ::      2 ::             6 ::$
$::   4 ::      3 ::            24 ::$
$::   5 ::      5 ::           120 ::$
$::   6 ::      8 ::           720 ::$
$::   7 ::     13 ::          5040 ::$
$::   8 ::     21 ::         40320 ::$
$::   9 ::     34 ::        362880 ::$
$::  10 ::     55 ::       3628800 ::$
$::  11 ::     89 ::      39916800 ::$
$::  12 ::    144 ::     479001600 ::$
$::  13 ::    233 ::    6227020800 ::$
$::  14 ::    377 ::   87178291200 ::$
$::  15 ::    610 :: 1307674368000 ::$
$:: ... ::    ... ::           ... ::$
```

There are some nice Unicode characters to make borders and separators with; see the Wikipedia page for box-drawing characters for example.

Given the numerical data (as an $r \times c$ table), as well as the header names, the separator string and the border string, output the tabulated data in the above form. Each data column should be right-aligned and represented by as few columns as possible, but there should be a column of spaces separating it from the separator or border strings.

Task Details

Your task is to implement a function called `tabulate`. This function has two positional arguments:

- the first is a `tuple`/`list` of c `str`s denoting the header names.
- a `tuple`/`list` of r `tuple`s/`list`s, each of which is composed of c `int`s or `str`s representing data.

In addition, it has three *keyword* arguments:

- the first is `sep`, a `str` denoting the separator string, which must default to `|`.
- the first is `lborder`, a `str` denoting the left border string, which must default to `|`.
- the first is `rborder`, a `str` denoting the right border string, which must default to `|`.

The function must `print` $r+1$ lines of output denoting the table. Do not print leading or trailing whitespace. It should not return anything.

Restrictions

(See 10a for more restrictions)

For this problem in particular:

- The following symbols are allowed: `map`, `filter`.
- The following import is allowed: `cache` and `lru_cache` from `functools`.
- The source code limit is 2000.

Example Calls

Example 1 Function Call

```
tabulate(('n', 'fib(n)', 'factorial(n)'), [
    [0, 0, 1],
    [1, 1, 1],
    [2, 1, 2],
    [3, 2, 6],
    [4, 3, 24],
    [5, 5, 120],
    [6, 8, 720],
    [7, 13, 5040],
    [8, 21, 40320],
    [9, 34, 362880],
    [10, 55, 3628800],
    [11, 89, 39916800],
    [12, 144, 479001600],
    [13, 233, 6227020800],
    [14, 377, 87178291200],
    [15, 610, 1307674368000],
])
```

Example 1 Output

```
|  n | fib(n) |  factorial(n) |
|  0 |      0 |             1 |
|  1 |      1 |             1 |
|  2 |      1 |             2 |
|  3 |      2 |             6 |
|  4 |      3 |            24 |
|  5 |      5 |           120 |
|  6 |      8 |           720 |
|  7 |     13 |          5040 |
|  8 |     21 |         40320 |
|  9 |     34 |        362880 |
| 10 |     55 |       3628800 |
| 11 |     89 |      39916800 |
| 12 |    144 |     479001600 |
| 13 |    233 |    6227020800 |
| 14 |    377 |   87178291200 |
| 15 |    610 | 1307674368000 |
```

Example 2 Function Call

```
tabulate(['n', 'fib(n)', 'factorial(n)'], (
    [0, 0, 1],
    [1, 1, 1],
    [2, 1, 2],
    [3, 2, 6],
    [4, 3, 24],
    [5, 5, 120],
    [6, 8, 720],
    [7, 13, 5040],
    [8, 21, 40320],
    [9, 34, 362880],
    [10, 55, 3628800],
    [11, 89, 39916800],
    [12, 144, 479001600],
    [13, 233, 6227020800],
    [14, 377, 87178291200],
    [15, 610, 1307674368000],
    ['...', '...', '...'],
), lborder='$::', rborder='::$', sep='::')
```

Example 2 Output

```
$::   n :: fib(n) ::  factorial(n) ::$
$::   0 ::      0 ::             1 ::$
$::   1 ::      1 ::             1 ::$
$::   2 ::      1 ::             2 ::$
$::   3 ::      2 ::             6 ::$
$::   4 ::      3 ::            24 ::$
$::   5 ::      5 ::           120 ::$
$::   6 ::      8 ::           720 ::$
$::   7 ::     13 ::          5040 ::$
$::   8 ::     21 ::         40320 ::$
$::   9 ::     34 ::        362880 ::$
$::  10 ::     55 ::       3628800 ::$
$::  11 ::     89 ::      39916800 ::$
$::  12 ::    144 ::     479001600 ::$
$::  13 ::    233 ::    6227020800 ::$
$::  14 ::    377 ::   87178291200 ::$
$::  15 ::    610 :: 1307674368000 ::$
$:: ... ::    ... ::           ... ::$
```

Constraints

- The function `tabulate` will be called at most 20 times.
- $1 \le r, c \le 40$
- Each header string will consist of at most 20 characters.
- Each string in a data cell will consist of at most 20 characters.
- Each separator and border string will consist of at most 5 characters.
- Each integer in a data cell will have at most 20 digits.
- All string characters will have ASCII values between 32 and 126, inclusive.

Scoring

- You get 120 ❤ points if you solve all test cases where:
  - The `sep`, `lborder` and `rborder` keywords are not used/passed.
- You get 60 ❤ points if you solve all test cases.

Submit solution

[CS 11] Practice 10 ❤

My submissions ❤

Points: 180 (partial)
Time limit: 3.0s
Memory limit: 1G

Author: kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3

Clarifications

Report an issue

No clarifications have been made at this time.