English

Week 6 - Scor Final: 0.8 Puncte

Create a python module (file name - *s6.py*) Implement the solutions for this test in this module.

Barem

Problem 1 = 1 points
Problem 2 = 1 points
Problem 3 = 1 points
Problem 4 = 1 points
Problem 5 = 1 points
Problem 6 = 1 points
Problem 7 = 1 points
Total = 7 points

Problem 1 - Total: 0.8 Puncte

Write a function named *fibo* that has a single parameter *n* (a positive interger). The function should return the *n*-th number of the Fibonnaci sequence (1, 1, 2, 3, 5, ...). The Fibonacci sequence is computed as follows: F(n) = F(n - 1) + F(n - 2), where F(1) = 1 and F(2) = 1.

Test #	Input	Output	Expected	Passed	Score
0	1	1	1	✓	1	0.00
1	4	3	3	✓	1	0.20
2	10	55	55	✓	1	0.20
3	**n** = 25	75025	75025	✓	1	0.20
4	**n** = 35	9227465	9227465	✓	1	0.20
5	**n** = 100	354224848179261111683542248481792619150	75	✗	0	0.00

Problem 2 - Total: 0 Puncte

Write a function named *is_prime* that has a single parameter *number* (a natural number). The function should return *True* if *number* is prime and *False* otherwise. A prime number is an integer greater than 1 that only has two factors: 1 and itself.

Test #	Input	Output	Expected	Passed	Score
0	1	Import Error	False	✗	0	0.00
1	2	Import Error	True	✗	0	0.00
2	15	Import Error	False	✗	0	0.00
3	29	Import Error	True	✗	0	0.00
4	**number** = 1337	Import Error	False	✗	0	0.00
5	**number** = 100003	Import Error	True	✗	0	0.00
6	**number** = 683713371921	Import Error	False	✗	0	0.00
7	**number** = 1713371921	Import Error	True	✗	0	0.00
8	**number** = 88133719213129371	Import Error	False	✗	0	0.00

Problem 3 - Total: 0 Puncte

Write a function named *custom_filter* that receives a list of natural numbers named *my_list*. The function should return a list of numbers sorted in ascending order from *my_list* that are both prime numbers and are part of the Fibonacci sequence. Hint: you can call previously defined functions.

Test #	Input	Output	Expected	Passed	Score
0	[2, 1, 4, 3]	Import Error	[2, 3]	✗	0	0.00
1	[29, 13, 5, 18, 21, 2, 9]	Import Error	[2, 5, 13]	✗	0	0.00
2	[12, 13, 8, 31, 11, 2, 29, 1, 5]	Import Error	[2, 5, 13]	✗	0	0.00

3	**my_list** = [610, 377, 987, 4181, 2584, 89, 144, 1597, 55, 6765, 233]	Import Error	[89, 233, 1597]	✗ 0	0.00
4	**my_list** = [877, 463, 233, 886, 211, 853, 349, 193, 907, 626, 199, 151, 641, 389, 487, 281, 766, 241, 358, 103, 379, 838, 634, 113]	Import Error	[233]	✗ 0	0.00
5	**my_list** = [17711, 121393, 832040, 46368, 28657, 514229, 317811, 6765, 10946, 196418, 75025]	Import Error	[28657, 514229]	✗ 0	0.00

Problem 4 - Total: 0 Puncte

Write a function named *extract_numbers* that receives a single string paramter named text. The function should return a list of numbers extracted from *text* sorted in descending order. A number in a string is defined as a contiguous sequence of digits. *text* won't contain numbers starting with 0.

Test #InputOutputExpectedPassedScore

0	"test12z34p5"	Import Error	[34, 12, 5]	✗ 0	0.00
1	"wan131NOaFZUMAATReke30E"	Import Error	[131, 30]	✗ 0	0.00
2	"R24SwT6Bl2r18c68ytwYLk125NkBgh943qTgkfp10jJoU"	Import Error	[943, 125, 68, 24, 18, 10, 6, 2]	✗ 0	0.00
3	**text** = "2DENDp4rD69THCZfU3mKFFoA2IsOphdrtqbYwDIvye7XHAvQ1n"	Import Error	[69, 7, 4, 3, 2, 2, 1]	✗ 0	0.00
4	**text** = "YTnWxNA1Tmh622od2I10dGHquqOuzaUaCmdSfJzQQ6RNt3dL6X532eF6MbYPQfZe6S61eUemd6cJnUwiiI"	Import Error	[622, 532, 61, 10, 6, 6, 6, 6, 6, 3, 2, 1]	✗ 0	0.00

Problem 5 - Total: 0 Puncte

Write a function named *special_sum* that receives a variable number of string parameters. The function should return the sum of the highest even number extracted from each text. Each string has at least one even number. Hint: use function *extract_numbers* from the previous problem.

Test #InputOutputExpectedPassedScore

0	"a2c3","b1d4"	Import Error	6	✗ 0	0
1	"a2b5c4","x8y9z10"	Import Error	14	✗ 0	0
2	"tPt82eWq31d10P","Lkn111N6aRekJ30E","90n32k1L8dnBa33"	Import Error	202	✗ 0	0
3	"R24SwT6Bl2r","18c68ytwYLk125","NkBgh943qTgkfp10jJoU","2DENDp4rD69THCZfU3mKFFoA","rtqbYwDIv77ye72XH"	Import Error	178	✗ 0	0
4	"NkBgh9R24Sc68ytwYLk125wT6Bl2r1843qTgk100DIv777yefprD69TH10jJoU12D3mKFFoArtqbYwENDp4CZfU72XH"	Import Error	100	✗ 0	0

Problem 6 - Total: 0 Puncte

Write a function named *loop* that receives a single dict parameter named *mapping*. This dictionary always contains a string key "start". Starting with the value of this key you must obtain a list of objects by iterating over *mapping* in the following way: the value of the current key is the key for the next value, until you find a loop (a key that was visited before). The function must return the list of objects obtained as previously described.

Test #InputOutputExpectedPassedScore

0	{'start': '1', '1': 'start'}	Import Error	['1']	✗ 0	0.00
1	{'start': '3', '3': '1', '1': '2', '2': 'start'}	Import Error	['3', '1', '2']	✗ 0	0.00

2	{'start': 'a', 'b': 'a', 'a': '6', '6': 'z', 'x': '2', 'z': '2', '2': '2', 'y': 'start'}	Import Error	['a', '6', 'z', '2']	✗	0	0.00
3	**mapping** = {'start': 'a', 'a': 'e', 'e': 'b', 'b': 'c', 'c': 'd', 'd': 'f', 'f': 'g', 'g': 'b'}	Import Error	['a', 'e', 'b', 'c', 'd', 'f', 'g']	✗	0	0.00
4	**mapping** = {'start': 'a', 'test': 'start', 'a': '6', 't': '2', 'python': 'test', '1': 'start', '6': 'f', 'f': 't', '2': 'python'}	Import Error	['a', '6', 'f', 't', '2', 'python', 'test']	✗	0	0.00

Problem 7 - Total: 0 Puncte

Write a function named *sequence* that receives a single natural number parameter named *n*. The function should return the *n*-th number in the sequence generated as follows: seq(n) = 2 * seq(int(n/2)) - seq(n-2), where seq(0) = 2 and seq(1) = 4.

Test #	Input	Output	Expected	Passed	Score
0	2	Import Error	6	✗	0 0.00
1	8	Import Error	10	✗	0 0.00
2	16	Import Error	18	✗	0 0.00
3	**n** = 44	Import Error	-2	✗	0 0.00
4	**n** = 97	Import Error	-28	✗	0 0.00
5	**n** = 7617	Import Error	-60	✗	0 0.00

Choose File no file selected Choose File... Token Submit

Upload Successful!