

Python for Math and Stat Fall 2021

Exam 1

1. (55 pts) For the following 5 problems, write down what each code block would display if executed in a Jupyter cell.

- (a) `(2021 % 100, 2021 // 10)`
- (b) `alist = list(range(9, 3, -2))
alist + [2 * alist[1]]`
- (c) `blist = [16, 26, 36, 46, 56, 66]
print(blist[3:4])
print(blist[4:])
print(blist[:-3])`
- (d) `clist = [1.5, 3.7, -2.8]
[x > 2 for x in clist]`
- (e) `lst = []
for n in range(4):
 print(n, lst)
 if n < 2:
 lst.append(n + 100)`

Solution:

- (a) `(21, 202)`
- (b) `[9, 7, 5, 14]`
- (c) `[46]
[56, 66]
[16, 26, 36]`
- (d) `[False, True, False]`
- (e) `0 []
1 [100]
2 [100, 101]
3 [100, 101]`

2. (15 pts) Write a function `even_odd_sums(nums)` that takes a list of ints and returns a 2-element tuple containing the sum of the even numbers in the list, followed by the sum of the odd numbers.

Example:

`even_odd_sums([-5, 18, 12, -3, -1])` returns (30, -9)

Solution:

```
def even_odd_sums(nums):
    even_sum = 0
    odd_sum = 0

    for n in nums:
        if n % 2 == 0:
            even_sum += n
        else:
            odd_sum += n

    return (even_sum, odd_sum)
```

OR

```
def even_odd_sums(nums):
    even_sum = sum(n for n in nums if n % 2 == 0)
    odd_sum = sum(n for n in nums if n % 2 == 1)

    return (even_sum, odd_sum)
```

3. (15 pts) Write a function `first_last(words)` that takes a list of non-empty strings and returns a new list with each string's first character copied to the end of the string.

Example:

`first_last(['ant', 'flu', 'x', 'suit'])`
returns ['anta', 'fluf', 'xx', 'suits']

Solution:

```
def first_last(words):
    return [w + w[0] for w in words]
```

4. (15 pts)

- (a) Write a function `str4(word)` that returns the first 4 letters of a non-empty string `word`. If `word` has fewer than 4 letters, pad with an 'x' to extend the length to 4 characters.

Examples:

`str4('university')` returns 'univ'

`str4('pa')` returns 'paxx'

- (b) Write a function `idkey(name, phone)` that combines the first 4 letters of `name` and the last 4 digits of `phone` to create an `identikey` in the form of a string. Assume that `name` is a non-empty string and that `phone` is a 10-digit int. If `name` has fewer than 4 letters, pad with an 'x'. This function should call `str4()`.

Examples:

`idkey('buffalo', 3034923456)` returns 'buff3456'

`idkey('vi', 7205551234)` returns 'vi xx1234'

Solution:

- ```
(a) def str4(word):
 wordlen = len(word)

 if wordlen < 4:
 return word + (4-wordlen) * 'x'
 else:
 return word[:4]

(b) def idkey(name, phone):
 return str4(name) + str(phone % 10000)
```

OR

```
def idkey(name, phone):
 return str4(name) + str(phone)[-4:]
```