

Assignment 22 Solutions

1. Create a function that takes three parameters where:

x is the start of the range (inclusive).

y is the end of the range (inclusive).

n is the divisor to be checked against.

Return an ordered list with numbers in the range that are divisible by the third parameter n. Return an empty list if there are no numbers that are divisible by n. **Examples:**

`list_operation(1, 10, 3)` → [3, 6, 9]

`list_operation(7, 9, 2)` → [8]

`list_operation(15, 20, 7)` → []

In [3]:

```
1 def list_operation(x,y,n):
2
3     divisor = []
4     nonDiv = []
5     for i in range(x,y+1):
6         if i % n == 0:
7             divisor.append(i)
8     return divisor
9 list_operation(1, 10, 3)
```

Out[3]:

[3, 6, 9]

In [4]:

```
1 list_operation(7, 9, 2)
```

Out[4]:

[8]

In [5]:

```
1 list_operation(15, 20, 7)
```

Out[5]:

[]

2. Create a function that takes in two lists and returns True if the second list follows the first list by one element, and False otherwise. In other words, determine if the second list is the first list shifted to the right by 1.

Examples:

```
simon_says([1, 2], [5, 1]) → True  
simon_says([1, 2], [5, 5]) → False  
simon_says([1, 2, 3, 4, 5], [0, 1, 2, 3, 4]) → True  
simon_says([1, 2, 3, 4, 5], [5, 5, 1, 2, 3]) → False
```

Notes:

1. Both input lists will be of the same length, and will have a minimum length of 2.
2. The values of the 0-indexed element in the second list and the n-1th indexed element in the first list do not matter.

In [6]:

```
1 def simon_says(ls,ls2):  
2     if ls[0:len(ls)-1] == ls2[1:len(ls2)]:  
3         return True  
4     return False  
5 simon_says([1, 2], [5, 1])  
6
```

Out[6]:

True

In [7]:

```
1 simon_says([1, 2], [5, 5])
```

Out[7]:

False

In [8]:

```
1 simon_says([1, 2, 3, 4, 5], [0, 1, 2, 3, 4])
```

Out[8]:

True

In [9]:

```
1 simon_says([1, 2, 3, 4, 5], [5, 5, 1, 2, 3])
```

Out[9]:

False

3.A group of friends have decided to start a secret society. The name will be the first letter of each of their names, sorted in a alphabetical order ? Create a function that takes in a list of names and returns the name of the secret society ?

Examples:

```
society_name(["Adam", "Sarah", "Malcolm"]) → "AMS"  
society_name(["Harry", "Newt", "Luna", "Cho"]) → "CHLN"
```

```
society_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"])
```

In [11]:

```
1 def society_name(lst):
2     return ''.join([i[0] for i in sorted(lst)])
3
4 society_name(["Adam", "Sarah", "Malcolm"])
```

Out[11]:

'AMS'

In [13]:

```
1 society_name(["Harry", "Newt", "Luna", "Cho"])
```

Out[13]:

'CHLN'

In [14]:

```
1 society_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"])
```

Out[14]:

'CJMPRR'

4. An isogram is a word that has no duplicate letters. Create a function that takes a string and returns either True or False depending on whether or not it's an "isogram".

Examples:

is_isogram("Algorism") → True

is_isogram("PasSword") → False

Not case sensitive.

is_isogram("Consecutive") → False

Notes:

Ignore letter case (should not be case sensitive).

All test cases contain valid one word strings.

In [15]:

```
1 def is_isogram(s):
2     for i in range(len(s)):
3         if s.lower().count(s[i]) > 1:
4             return False
5     else:
6         return True
7
8 is_isogram("Algorism")
```

Out[15]:

True

In [16]:

```
1 is_isogram("PasSword")
```

Out[16]:

False

In [17]:

```
1 is_isogram("Consecutive")
```

Out[17]:

False

5. Create a function that takes a string and returns True or False, depending on whether the characters are in order or not ?

Examples:

```
is_in_order("abc") → True
is_in_order("edabit") → False
is_in_order("123") → True
is_in_order("xyzz") → True
```

Notes:

You don't have to handle empty strings.

In [19]:

```
1 def is_in_order(s):  
2     n = len(s)  
3     c = [s[i] for i in range(len(s))]  
4     c.sort()  
5     for i in range(n):  
6         if c[i] != s[i]:  
7             return False  
8     return True  
9  
10 is_in_order("abc")
```

Out[19]:

True

In [20]:

```
1 is_in_order("edabit")
```

Out[20]:

False

In [21]:

```
1 is_in_order("123")
```

Out[21]:

True

In [22]:

```
1 is_in_order("xyzz")
```

Out[22]:

True