# Python: Exercise on Lists and Strings

**Name: Mark Lopes**

**Roll No: 9913**

**S.E Comps_A Batch C**

Q1. Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last chars of the string are the same.

```
def count_string(list):
    count = 0
    for string in list :
        if len(string)>=2 and string[0]==string[-1]:
            count += 1
    return count


list1=["hello","mam","u","yuy","hf fh"]
final_count=count_string(list1)
print("The count is:")
print(final_count)
```

```
"C:\Users\Mark Lopes\PycharmProjects\pytho
The count is:
3

Process finished with exit code 0
```

Q2. Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first.

```
def string_sort(strings_list):
    # Initialize empty lists
    starts_with_x = []
    does_not_start_with_x = []

    # Separate the strings into two groups
    for string in strings_list:
        if string and string[0] == 'x':
            starts_with_x.append(string)
        else:
            does_not_start_with_x.append(string)

    # Sort both groups
    starts_with_x.sort()
    does_not_start_with_x.sort()

    # Combine the sorted lists
    result = starts_with_x + does_not_start_with_x

    return result


list=["xampp","pycharm","vs_code","xpfs","area"]
sorted_list = string_sort(list)
print(sorted_list)
```

```
"C:\Users\Mark Lopes\PycharmProjects\pythonProject\.v
['xampp', 'xpfs', 'area', 'pycharm', 'vs_code']

Process finished with exit code 0
```

Q3. Given two lists sorted in increasing order, create and return a merged list of all the elements in sorted order. You may modify the passed in lists. Ideally, the solution should work in "linear" time, making a single pass of both lists.

```
def sort_both(s1,s2):
    final_sort=s1+s2
    final_sort=sorted(final_sort)
    return final_sort


list1 = ["d","t","c"]
list2 = ["z","b","a"]

print(sort_both(list1,list2))
```

```
"C:\Users\Mark Lopes\PycharmProjects\pyt
['a', 'b', 'c', 'd', 't', 'z']

Process finished with exit code 0
```

Q4. Given a string s, return a string made of the first 2 and the last 2 chars of the original string, so 'spring' yields 'spng'. However, if the string length is less than 2, return instead the empty string.

```
def q4(s):
    if len(s)<2:
        return ""
    string = s[:2] + s[-2:]
    return string


test_string="Mark Lopes"
print(q4(test_string))
```

```
"C:\Users\Mark Lopes\PycharmProjects\p
Maes

Process finished with exit code 0
```

Q5. Given a string s, return a string where all occurences of its first char have been changed to '*', except do not change the first char itself.

```
test_string = "engineering"
print(test_string[0]+test_string[1:].replace(test_string[0],'*'))
```

```
"C:\Users\Mark Lopes\PycharmProjects\
engin**ring

Process finished with exit code 0
```

Q6. Given strings a and b, return a single string with a and b separated by a space '<a> <b>', except swap the first 2 chars of each string.

```
s1 = input("Enter first string\n")
s2 = input("Enter second string\n")

print(s2[0]+s1[1:]+" "+s1[0]+s2[1:])
```

```
"C:\Users\Mark Lopes\Pyc
Enter first string
Hello
Enter second string
World
Wello Horld
```

Q7. Given a string, find the first appearance of the substring 'not' and 'bad'. If the 'bad' follows the 'not' replace the whole 'not'...'bad' substring with 'good'. Return the resulting string.

```
def check_string(s):
    if (s.find('not')!=-1 and s.find('bad')!=-1):
        print(s[:s.find('not')]+"good")


test_string=input("Enter first string: ")
check_string(test_string)
```

```
"C:\Users\Mark Lopes\PycharmProjects\pythonProject\.
Enter first string: The weather is not bad
The weather is good

Process finished with exit code 0
```