

Data Structure – Exercise

I. Câu hỏi tự luận

1. Cho một list các số nguyên `num_list` và một sliding window có kích thước size `k` di chuyển từ trái sang phải. Mỗi lần dịch chuyển 1 vị trí sang phải có thể nhìn thấy được `k` số trong `num_list` và tìm số lớn nhất trong `k` số này sau mỗi lần trượt `k` phải lớn hơn hoặc bằng 1

Input: `num_list = [3, 4, 5, 1, -44, 5, 10, 12, 33, 1]`

Output: `[5, 5, 5, 5, 10, 12, 33, 33]`

Ví dụ:

`[3, 4, 5], 1, -44, 5, 10, 12, 33, 1 => max 5`
`3, [4, 5, 1], -44, 5, 10, 12, 33, 1 => max 5`
`3, 4, [5, 1, -44], 5, 10, 12, 33, 1 => max 5`
`3, 4, 5, [1, -44, 5], 10, 12, 33, 1 => max 5`
`3, 4, 5, 1, [-44, 5, 10], 12, 33, 1 => max 10`
`3, 4, 5, 1, -44, [5, 10, 12], 33, 1 => max 12`
`3, 4, 5, 1, -44, 5, [10, 12, 33], 1 => max 33`
`3, 4, 5, 1, -44, 5, 10, [12, 33, 1] => max 33`

2. Cho hai list các số nguyên là `num_list1` và `num_list2`, trả về các list các số cũng xuất hiện trong hai list, thứ tự không quan trọng.

Input: `nums1 = [1,2,2,1], nums2 = [2,2]`

Output: `[2,2]`

Input: `nums1 = [4,9,5], nums2 = [9,4,9,8,4]`

Output: `[4,9]` or `[9, 4]`

3. Thực hiện theo các yêu cầu sau.

Viết function trả về một dictionary đếm số lượng chữ xuất hiện trong một từ, với key là chữ cái và value là số lần xuất hiện

- **Input:** một từ
- **Output:** dictionary đếm số lần các chữ xuất hiện
- **Note:** Giả sử các từ nhập vào đều có các chữ cái thuộc [a-z] hoặc [A-Z]

```
1 # Examples 1(a)
2 string = 'Happiness'
3 count_chars(string)
4 >> {'H': 1, 'a': 1, 'e': 1, 'i': 1, 'n': 1, 'p': 2, 's': 2}
5
6 string = 'smiles'
7 count_chars(string)
8 >> {'e': 1, 'i': 1, 'l': 1, 'm': 1, 's': 2}
```

Code Listing 1: Đây là các ví dụ các bạn không cần thiết đặt tên giống ví dụ

4. Thực hiện theo các yêu cầu sau .

Viết function đọc các câu trong một file txt, đếm số lượng các từ xuất hiện và trả về một dictionary với key là từ và value là số lần từ đó xuất hiện.

- **Input:** Đường dẫn đến file txt
- **Output:** dictionary đếm số lần các từ xuất hiện
- **Note:**
 - Giả sử các từ trong file txt đều có các chữ cái thuộc [a-z] hoặc [A-Z]
 - Không cần các thao tác xử lý string phức tạp **nhưng cần xử lý các từ đều là viết thường**
 - Các bạn dùng lệnh này để download và tham khảo Code Listing 2
!gdown <https://drive.google.com/uc?id=1IBScGdW2xlNsc9v5zSAya548kNgiOrko>

```
1 # Examples 1(b)
2 !gdown https://drive.google.com/uc?id=1IBScGdW2xlNsc9v5zSAya548kNgiOrko
3 file_path = '/content/P1_data.txt'
4 word_count(file_path)
5 >>{'a': 7,
6   'again': 1,
7   'and': 1,
8   'are': 1,
9   'at': 1,
10  'be': 1,
11  'become': 2,
12  ...}
```

Code Listing 2: Đây là các ví dụ các bạn không cần thiết đặt tên giống ví dụ

II. Câu hỏi trắc nghiệm

- Đọc tự luận trước để nắm được idea tổng quát (sẽ không yêu cầu nhưng khuyến khích các bạn tự làm tự luận) và các bài này sẽ được giải trong buổi TA.
- Các bạn phải làm phần trắc nghiệm
 - Các câu hỏi có ký hiệu **(LT)**: là các câu hỏi lý thuyết và đã có sẵn trong file hint nên các bạn chỉ cần hiểu và chạy lại để chọn được đáp án đúng
 - Các câu hỏi có ký hiệu **(Code)**: là câu hỏi yêu cầu các bạn phải trực tiếp code vào phần bị khuyết để có thể chọn được đáp án đúng
 - **Lưu ý**: Đối với dạng câu hỏi **(Code)** trong file hint luôn có 1 test case bắt đầu với từ khóa assert nếu các bạn chạy không báo lỗi có nghĩa các bạn đã vượt qua được test case này và chạy lệnh tiếp theo để trả lời câu hỏi trắc nghiệm
 - **Lưu ý**: Đọc kỹ các code gợi ý và code ví dụ mẫu ở tự luận có thể sẽ có ích cho các bạn khi làm trắc nghiệm

Câu hỏi 1:(Code) Hoàn thành chương trình sau với mô tả bài toán từ câu I.1. Đầu ra của chương trình dưới đây là gì?

```
1 def max_in_window(num_list, k = 3):
2     # Your code here
3
4     # End your code
5
6 test = [3, 4, 5, 1, -44]
7 assert max_in_window(num_list = test, k = 3) == [5, 5, 5]
8
9 num_list = [3, 4, 5, 1, -44, 5, 10, 12, 33, 1]
10 print(max_in_window(num_list = num_list, k = 3))
```

- a) [5, 5, 5, 5, 10, 12, 33, 33]
- b) [2, 5, 3, 4, 1, 10, 3, 3]
- c) [0, 9, 5, 1, 0, 12, 3, 33]
- d) Raise an Error

Câu hỏi 2:(Code) Hoàn thành chương trình sau với mô tả bài toán từ câu I.2. Đầu ra của chương trình dưới đây là gì?

```
1 def intersection(lst1, lst2):
2     # Your code here
3
4     # End your code
5
6 test_list1 = [4, 9]
7 test_list2 = [9, 2]
8 assert intersection(lst1 = test_list1, lst2 = test_list2) == [9]
9
10 num_list1 = [4, 9, 5]
11 num_list2 = [9, 4, 9, 8, 4]
12 intersection(num_list1, num_list2)
```

- a) [9, 9]
- b) [4, 9]

- c) [4, 2]
d) Raise an Error

Câu hỏi 3:(Code) Hoàn thành chương trình sau với mô tả bài toán từ câu I.3. Đầu ra của chương trình dưới đây là gì?

```
1 def count_chars(inp_str):  
2     # Your code here  
3  
4     # End your code  
5  
6 test_string = 'Happiness'  
7 assert count_chars(inp_str = test_string) == {'H': 1, 'a': 1, 'p': 2, 'i': 1, 'n': 1,  
8     'e': 1, 's': 2}  
9  
10 string = 'smiles'  
11 count_chars(string)
```

- a) 's': 2, 'm': 1, 'i': 1, 'l': 1, 'e': 1
b) 's': 0, 'm': 1, 'i': 1, 'l': 1, 'e': 8
c) 's': 4, 'm': 1, 'i': 2, 'l': 1, 'e': 1
d) Raise an Error

Câu hỏi 4:(Code) Hoàn thành chương trình sau với mô tả bài toán từ câu I.4. Đầu ra của chương trình dưới đây là gì?

```
1 !gdown https://drive.google.com/uc?id=1IBScGdW2xlNsc9v5zSAya548kNgi0rko  
2  
3 def count_words(file_path):  
4     # Your code here  
5  
6     # End your code  
7  
8 file_path = 'P1_data.txt'  
9 assert count_words(file_path)['success'] == 3  
10  
11 file_path = 'P1_data.txt'  
12 count_words(file_path)['man']
```

- a) 4
b) 5
c) 6
d) 9

Câu hỏi 5:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 n = 1  
2 for i in range(0, 500, 100):  
3     n = i  
4 print(n)
```

- a) 900
b) 400
c) None
d) Raise an Error

Câu hỏi 6:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 data = "I'm learning Python!"  
2 print(data.split()[1])
```

- a) Python
- b) learning
- c) I'm
- d) Raise an Error

Câu hỏi 7:(Code) Hoàn thành chương trình sau. Đầu ra của chương trình dưới đây là gì?

```
1 def check_the_number(N):  
2     list_of_numbers = []  
3     result = ""  
4     for i in range(1, 5):  
5         #Your code here  
6         #Su dung append them i vao trong list_of_number  
7     if N in list_of_numbers:  
8         results = "True"  
9     if N not in list_of_numbers:  
10        results = "False"  
11    return results  
12  
13 N = 7  
14 assert check_the_number(N) == "False"  
15  
16 N = 2  
17 results = check_the_number(N)  
18 print(results)
```

- a) True
- b) False
- c) None
- d) Raise an Error

Câu hỏi 8:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 data = "my list : [ 1, 2, [3, 4]]"  
2 print(type(data))
```

- a) <class 'str'>
- b) <class 'list'>
- c) <class 'mydata'>
- d) Raise an Error

Câu hỏi 9:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 "Eiffel" > "Apple"
```

- a) True
- b) False
- c) Raise an Error
- d) None

Câu hỏi 10:(LT) Đầu ra của chương trình dưới đây là gì?

```

1 n = 0
2 for i in range(5):
3     n += i
4     if n>0 and n%3 == 0:
5         break
6 print(n)

```

- a) 0
- b) 3**
- c) Raise an Error
- d) 6

Câu hỏi 11:(LT) Đầu ra của chương trình dưới đây là gì?

```

1 space1 = "Cherry Blossom After Winter"
2 space2 = "Flowers are blooming on the hillsides, which signals the coming of spring"
3 space = space1 + space2
4 print(space[-6:])

```

- a) spring**
- b) Raise an Error
- c) Winter
- d) A and C

Câu hỏi 12:(LT) Đầu ra của chương trình dưới đây là gì?

```

1 my_list = [0, 1, 1, 2, 1]
2 odd = 1
3 even = 0
4
5 for number in my_list:
6     if number % 2 == 0:
7         odd += number
8     else:
9         even += number
10 print(f"odd:{odd}, even:{even}")

```

- a) odd:1, even:0**
- b) odd:3, even:10
- c) odd:3, even:3
- d) Raise an Error

Câu hỏi 13:(LT) Đầu ra của chương trình dưới đây là gì?

```

1 weather = '@the drizzle in spring makes the air more humid. @'
2 me = "i love iT!"
3
4 txt = weather.strip('@').capitalize()+ me.title()
5 print(txt)

```

- a) The drizzle in spring makes the air more humid. I Love It!**
- b) The drizzle in spring makes the air more humid. i love it!
- c) the drizzle in spring makes the air more humid. @@ I Love It!
- d) Raise an Error

Câu hỏi 14:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 data = [1, 9, 3, -3 ]
2 data.sort()
3 print(data)
```

- a) [-3, 1, 3, 9]
- b) [1, 2, 3, 4]
- c) [9, 3, 1, -3]
- d) Raise an Error

Câu hỏi 15:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 my_exam = "Under the drizzle, the flower field seems covering with the glitter water drops!"
2 print(my_exam.strip('!').split())
```

- a) ['Under', 'the', 'drizzle,', 'the', 'flower', 'field', 'seems', 'covering', 'with', 'the', 'glitter', 'water', 'drops']
- b) Raise an Error
- c) ['Under', 'the', 'drizzle,', 'the', 'flower', 'field', 'seems', 'covering', 'with', 'the', 'glitter', 'water', 'drops', '!']
- d) a, c

Câu hỏi 16:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 my_string = "Peach blossoms bloom in spring"
2
3 my_bag_of_word = []
4 for element in my_string:
5     if element not in my_bag_of_word:
6         my_bag_of_word.append(element)
7 print(my_bag_of_word)
```

- a) ['P', 'e', 'a', 'c', 'h']
- b) [' ']
- c) Raise an Error
- d) ['P', 'e', 'a', 'c', 'h', ' ', 'b', 'l', 'o', 's', 's', 'o', 'm', 's', 'i', 'n', 'p', 'r', 'i', 'n', 'g']

Câu hỏi 17:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 my_string = "My name is Tom"
2 results = my_string.count('m')
3
4 print(results)
```

- a) Raise an Error
- b) 2
- c) 3
- d) 4

Câu hỏi 18:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 even_numbers = [x for x in range(1, 5) if x % 2==0]
2 print(even_numbers)
```

- a) Raise an Error
- b) [1, 2, 3, 4, 5]
- c) [2, 4]
- d) [3, 5]

Câu hỏi 19:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 X = [[1, 1],
2      [2, 2]]
3
4 result = [[0,0],
5           [0,0]]
6
7 for i in range(len(X)):
8     for j in range(len(X[0])):
9         result[j][i] = X[i][j]
10
11 for r in result:
12     print(r)
```

- a) [1, 2]
[1, 2]
- b) [1, 2, 3] [1, 2, 3]
- c) [1, 2, 1, 2]
- d) Raise an Error

Câu hỏi 20:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(my_data):
2     rs = 0
3     for i in my_data:
4         rs = rs + i
5     return rs
6
7 my_list = [1, 2, 3]
8 print(my_function(my_data = my_list))
```

- a) 5
- b) 6
- c) 8
- d) 9

Câu hỏi 21:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(my_data):
2     result = []
3     for element in data:
4         if element not in result:
5             result.append(element)
6     return result
7
8 data = [ {'id': 'M12'}, 10, 20, 30]
9 print(my_function(data))
```

- a) [{'id': 'M12'}, 10, 20, 30]
- b) ['id', 'M12', 10, 20, 30]
- c) [10, 20, 30]
- d) Raise an Error

Câu hỏi 22:(Code) Hãy hoàn thành chương trình dưới đây. Đầu ra của chương trình là gì?

```
1 def my_function(data, max, min):
2     result = []
3     for i in data:
4         #Your code here
5         #Neu i < min thi them min vao result
6         elif i > max:
7             result.append(max)
8         else:
9             result.append(i)
10    return result
11 my_list = [5, 2, 5, 0, 1]
12 max = 1
13 min = 0
14 assert my_function(max = max, min = min, data = my_list) == [1, 1, 1, 0, 1]
15 my_list = [10, 2, 5, 0, 1]
16 max = 2
17 min = 1
18 print(my_function(max = max, min = min, data = my_list))
```

- a) [10, 2, 5, 1, 1]
- b) [0, 2, 2, 0, 0]
- c) [2, 2, 2, 1, 1]
- d) Raise an Error

Câu hỏi 23:(code) Hãy hoàn thành chương trình dưới đây. Đầu ra của chương trình là gì?

```
1 def my_function(x, y):
2     #Your code here
3     #Su dung extend de noi y vao x
4     #return x
5
6 list_num1 = ['a', 2, 5]
7 list_num2 = [1, 1]
8 list_num3 = [0, 0]
9
10 assert my_function(list_num1, my_function(list_num2, list_num3)) == ['a', 2, 5, 1, 1,
    0, 0]
11
12 list_num1 = [1, 2]
13 list_num2 = [3, 4]
14 list_num3 = [0, 0]
15
16 print(my_function(list_num1, my_function(list_num2, list_num3)))
```

- a) [1, 2, 3, 4, 0, 0]
- b) [1, 2, [3, 4, 0, 0]]
- c) [[1, 2, 3, 4, 0, 0]]
- d) Raise an Error

Câu hỏi 24:(code) Hãy hoàn thành chương trình tìm phần tử có giá trị nhỏ nhất trong một list dưới đây. Đầu ra của chương trình là gì?

```
1 def my_function(n):
2     #Your code here
3
4 my_list = [1, 22, 93, -100]
5 assert my_function(my_list) == -100
6
7 my_list = [1, 2, 3, -1]
8 print(my_function(my_list))
```

- a) None
- b) Raise an Error
- c) -1
- d) 3

Câu hỏi 25:(code) Hãy hoàn thành chương trình tìm phần tử có giá trị lớn nhất trong một list dưới đây. Đầu ra của chương trình là gì?

```
1 def my_function(n):
2     #Your code here
3
4 my_list = [1001, 9, 100, 0]
5 assert my_function(my_list) == 1001
6
7 my_list = [1, 9, 9, 0]
8 print(my_function(my_list))
```

- a) None
- b) Raise an Error
- c) 0
- d) 9

Câu hỏi 26:(code) Hãy hoàn thành chương trình dưới đây. Đầu ra của chương trình là gì?

```
1 def My_function(integers, number = 1):
2     return any(#Your code here: Thuc hien duyet tung phan tu trong integers, so sanh
3               tung phan tu voi number, neu bang nhau tra ve True, khac nhau tra ve false
4               #vi du: integers = [1, 2, 3], number = 2, ban se tao ra list [False,
5               True, False] )
6
7 my_list = [1, 3, 9, 4]
8 assert My_function(my_list, -1) == False
9
10 my_list = [1, 2, 3, 4]
11 print(My_function(my_list, 2))
```

- a) 1
- b) 4
- c) True
- d) False

Câu hỏi 27:(code) Hãy hoàn thành chương trình dưới đây. Đầu ra của chương trình là gì?

```
1 def my_function(list_nums = [0, 1, 2]):
2     var = 0
3     for i in list_nums:
4         var += i
5     return var/len(list_nums) #Tra ve gia tri trung binh cua list bang cach chia var cho
6                               so luong phan tu trong list_nums
7
8 assert my_function([4, 6, 8]) == 6
9 print(my_function())
```

- a) 1.0
- b) 2.0
- c) Raise an Error
- d) A and C

Câu hỏi 28:(LT) Đầu ra của chương trình là gì?

```
1 def my_function(signal1, signal2):
2     var = False
3     for s1 in signal1:
4         for s2 in signal2:
5             if s1 == s2:
6                 var = True
7                 return var
8 print(my_function([1, 1, 1], [2, 2, 2]))
```

- a) False
- b) True
- c) Raise an Error
- d) None

Câu hỏi 29:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(signal1, signal2):
2     var = False
3     for s1 in signal1:
4         for s2 in signal2:
5             if s1 == s2:
6                 var = True
7                 return var
8 print(my_function([1, 2, 3], [2, 2]))
```

- a) False
- b) True**
- c) Raise an Error
- d) None

Câu hỏi 30:(code) Hãy hoàn thành chương trình dưới đây. Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(data):
2     var = []
3     for i in data:
4         #Your code here if i% 3 == 0:
5         #Neu i chia het cho 3 thi them i vao list var
6         var.append(i)
7     return var
8 assert my_function([3, 9, 4, 5]) == [3, 9]
9 print(my_function([1, 2, 3, 5, 6]))
```

- a) [3, 6]**
- b) [1, 2, 3, 5, 6]
- c) a and d
- d) [5, 1]

Câu hỏi 31:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(x):
2     for i in range(x):
3         for j in range(x):
4             if i == j:
5                 print("1 ", end=" ")
6             else:
7                 print("0 ", end=" ")
8             print()
9
10 my_function(2)
```

- a)**
1 0
0 1
- b)
0 1
1 0
- c) [1, 0, 0, 1]
- d) None

Câu hỏi 32:(code) Hãy hoàn thành chương trình sau đây thực hiện tính giai thừa của 1 số. Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(y):
2     var = 1
3     while(y > 1):
4         #Your code here var *= y
5     return var        y -=1
6 assert my_function(8) == 40320
7 print(my_function(4))
```

- a) 0
- b) 20
- c) 24
- d) Raise an Error

Câu hỏi 33:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 def my_function(signal):
2     var = True
3     while var:
4         var = False
5         for i in range(len(signal) - 1):
6             if signal[i] > signal[i + 1]:
7                 signal[i], signal[i + 1] = signal[i + 1], signal[i]
8                 var = True
9
10 my_signal = [1, 2, 3, 0]
11 my_function(my_signal)
12 print(my_signal)
```

- a) [0, 1, 2, 3]
- b) [1, 1, 1, 1]
- c) [3, 2, 1, 0]
- d) Raise an Error

Câu hỏi 34:(LT) Đầu ra của chương trình dưới đây là gì?

```
1 def sum_function(num1, num2):
2     '''
3     Sum of two numbers
4     input: num1, num2
5     output: num1+num2
6     '''
7     return num1 + num1
8
9 print(sum_function.__doc__)
```

- a) Sum of two numbers
input: num1, num2
output: num1+num2
- b) 10
- c) Raise an Error
- d) 100!

Câu hỏi 35:(code) Hãy hoàn thành chương trình đảo ngược chuỗi dưới đây. Đầu ra của chương trình là gì?

```
1 def my_function(x):
2     #your code here
3     return x[::-1]
4 x = 'I can do it'
5 assert my_function(x)=="ti od nac I"
6
7 x = 'apricot'
8 print(my_function(x))
```

- a) apricot
- b) tocirpa
- c) Raise a Error
- d) None

Câu hỏi 36:(code) Hãy hoàn thành chương trình dưới đây. Đầu ra của chương trình là gì?

```
1 def function_helper(x):
2     #Your code here
3     #Neu x>0 tra ve 'T', nguoc lai tra ve 'N'
4     return 'T' if x > 0 else 'N'
5 def my_function(data):
6     res = [function_helper(x) for x in data]
7     return res
8
9 data = [10, 0, -10, -1]
10 assert my_function(data) == ['T', 'N', 'N', 'N']
11
12 data = [2, 3, 5, -1]
13 print(my_function(data))
```

- a) ['N', 'T', 'T', 'N']
- b) ['T', 'N', 'T', 'N']
- c) ['T', 'T', 'T', 'N']
- d) Raise an Error

Câu hỏi 37:(code) Hãy hoàn thành chương trình dưới đây để loại bỏ những phần tử trùng nhau. Đầu ra của chương trình là gì?

```
1 def function_helper(x, data):
2     for i in data:
3         #Your code here     if x == i:
4         #Neu x == i thi return 0 return 0
5     return 1
6
7 def my_function(data):
8     res = []
9     for i in data:
10        if function_helper(i, res):
11            res.append(i)
12
13    return res
14
15 lst = [10, 10, 9, 7, 7]
16 assert my_function(lst)==[10, 9, 7]
17
18 lst = [9, 9, 8, 1, 1]
19 print(my_function(lst))
```

a) [9, 8, 1]

b) [1, 1, 1]

c) [9, 9, 8, 1, 1]

d) Raise an Error