

## Chapter 4 - Exercise 1: Thực hiện những yêu cầu liên quan đến series

### Part 1: Thực hiện các phép toán trên series

```
import numpy as np
import pandas as pd
```

```
# Câu 1a: Cho arr_1 là mảng số nguyên chẵn [2, 4, 6, 8, 10], arr_2 là mảng số ng
# Tạo biến kiểu Serries ser1 từ arr_1, ser2 từ arr_2
arr_1 = np.array([2, 4, 6, 8, 10])
arr_2 = np.array([1, 3, 5, 7, 11])
ser1 = pd.Series(arr_1)
ser2 = pd.Series(arr_2)
# In danh sách các phần tử của ser1 và ser2
print(ser1)
print(ser2)
```

```
0      2
1      4
2      6
3      8
4     10
dtype: int32
0      1
1      3
2      5
3      7
4     11
dtype: int32
```

```
# Câu 1b: Thực hiện phép toán và thể hiện kết quả của: ser1 + ser2
ser1 + ser2
```

```
0      3
1      7
2     11
3     15
4     21
dtype: int32
```

```
# Câu 1c: Thực hiện phép toán và thể hiện kết quả của: ser1 - ser2
ser1 - ser2
```

```
0      1
1      1
2      1
```

```
0      2
1     12
2     30
3     56
4    110
dtype: int32
```

```
# Câu 1e: Thực hiện phép toán và thể hiện kết quả của: ser1 / ser2
ser1 / ser2
```

```
0      2.000000
1      1.333333
2      1.200000
3      1.142857
4      0.909091
dtype: float64
```

```
# Câu 2a: Kiểm tra xem các phần tử của ser1 có > các phần tử của ser2 không?
ser1 > ser2
```

```
0      True
1      True
2      True
3      True
4     False
dtype: bool
```

```
# Câu 2b: Kiểm tra xem các phần tử của ser1 có < các phần tử của ser2 không?
ser1 < ser2
```

```
0     False
1     False
2     False
3     False
4       True
dtype: bool
```

```
# Câu 2c: Kiểm tra xem các phần tử của ser1 có = các phần tử của ser2 không?
ser1 == ser2
```

```
0     False
1     False
2     False
3     False
4     False
dtype: bool
```

```
dtype: int64
```

```
# Câu 3b: Tạo series ser3 chỉ chứa các phần tử có trong ser1 mà không có trong ser2
#          In danh sách các phần tử của ser3
ser3 = ser1[~ser1.isin(ser2)]
print(ser3)
```

```
0      2
1      4
3      8
4     10
dtype: int32
```

```
# Câu 3c: Tạo series ser4 chỉ chứa các phần tử có trong ser2 mà không có trong ser1
#          In danh sách các phần tử của ser4
ser4 = ser2[~ser2.isin(ser1)]
print(ser4)
```

```
0      1
1      3
2      5
3      7
4     11
1     12
dtype: int64
```

```
# Câu 4: Tạo series ser5 chứa các phần tử chỉ có trong ser1 và chỉ có trong ser2
#          In danh sách các phần tử của ser5
ser5 = ser3.append(ser4)
print(ser5)
```

```
0      2
1      4
3      8
4     10
0      1
1      3
2      5
3      7
4     11
1     12
dtype: int64
```

```
1      9
2      6
3      1
4      1
dtype: int32
30     8
31     4
32     7
33     6
34     2
dtype: int32
```

```
# Câu 1b: In danh sách các phần tử của ser6 theo dạng array
print(ser6.values)
```

```
[6 9 6 1 1 2 8 7 3 5 6 3 5 3 5 8 8 2 8 1 7 8 7 2 1 2 9 9 4 9 8 4 7 6 2]
```

```
# Câu 1c: Cho biết thông tin thống kê chung (describe()) của ser6
ser6.describe()
```

```
count      35.000000
mean        5.200000
std         2.763204
min         1.000000
25%         2.500000
50%         6.000000
75%         8.000000
max         9.000000
dtype: float64
```

```
# Câu 1d: Cho biết tổng của các phần tử có trong ser6
ser6.sum()
```

```
182
```

```
# Câu 4: In ra các giá trị unique (array) trong ser6
ser6.unique()
```

```
array([6, 9, 1, 2, 8, 7, 3, 5, 4], dtype=int64)
```

```
# Câu 5: Tạo series ser7 với mỗi phần tử có giá trị = lập phương của phần tử trong ser6
import math
ser7 = ser6.map(lambda x: math.pow(x,3))

# Xem 5 dòng dữ liệu đầu tiên (head) của ser7
print(ser7.head())
```

```
0    216.0
1    729.0
2    216.0
3         1.0
4         1.0
dtype: float64
```

## Part 3: Tạo series từ list, chuỗi và biểu thức điều kiện

```
# Câu 1: Cho list sau:
lst = ["abc", "defg", "htmlmj", "dfg", "ljsac"]
```

```
        count += 1
    return count == 2

ser_nguyento = ser.map(lambda x: test_prime(x))

print(ser[ser_nguyento])
```

```
1    2
3    5
5    7
dtype: int32
```

```
# Câu 3: Cho mẫu email như sau:
pattern = '[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}'

# Tạo một series ser_ch, với mỗi phần tử trong ser_ch là một chuỗi
# Gợi ý: 'reading newspaper from tuoitre.vn', 'tubirona@gmail.com', 'nguyen.nn@y:
ser_ch = pd.Series(['reading newspaper from tuoitre.vn', 'tubirona@gmail.com', 'n

# In ra những dòng trong ser_ch thỏa điều kiện chuỗi là email
import re
is_email = ser_ch.map(lambda x: bool(re.match(pattern, x)))
```

