1. Hãy tìm số lớn nhất trong 3 số a,b,c bất kỳ nhập vào

2. Cho 2 số a,b hãy đổi giá trị của 2 số cho nhau (theo 2 cách: 1 xài biến tạm, 2 không xài biến tạm)

Ví dụ:

Input: a = 1, b = 2

Output: a = 2, b = 1

Cách tạo biến tạm:

temp = a

a = b

b = temp

3. Nhập vào 1 số bất kỳ, kiểm tra số đó có phải số nguyên tố hay không

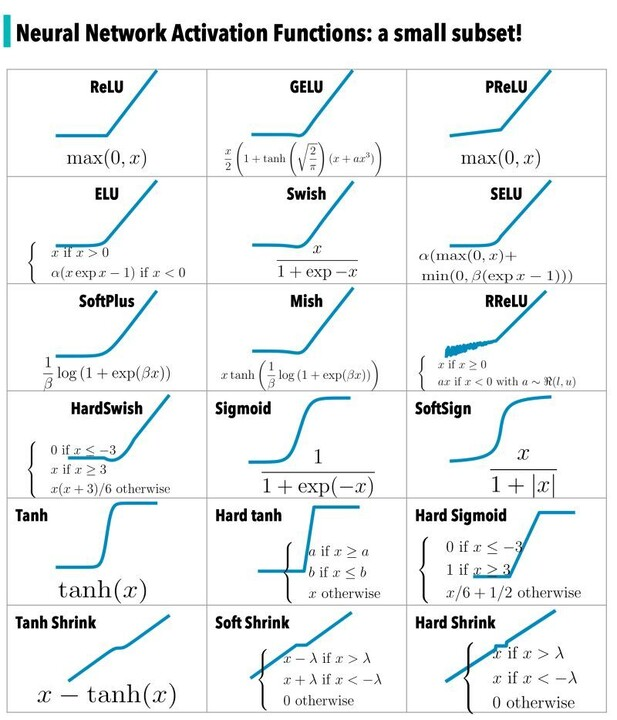
4. Cho số bất kỳ từ 2 chữ số trở lên, hãy thực hiện tính tổng các chữ số trong số đó tới khi thu dc 1 số có 1 chữ số

Ví dụ: 129 = 1 + 2 + 9 = 12 = 1 + 2 = 3

5. Cho list sau: lst\_data = [10, 1, 2, 5, 3, 6, 8, 20, 50], hãy sắp xếp list theo thứ tự tăng dần (không dùng hàm có sẵn)

6. Cho list sau: lst\_data = [1, 2, 3, 0, -1, 2, 3, 1].

Hãy cài đặt các hàm ReLU, sigmoid, tanh, LeakyReLU cho lst\_data



7. Cho list sau: lst\_data = [1, 2, 5, 0, 7, -1]. Hãy tạo ra list với gtri theo quy luật sau:

+ giá trị đầu tiên của list mới = giá trị đầu tiên của list cũ

+ giá trị tiếp theo bằng giá trị kế tiếp list cũ + giá trị phía trước list cũ

Ví dụ:

+ Input: lst\_data = [1, 2, 5, 0, 7, -1]

+ Output: lst\_output = [1, 3, 7, 5, 7, 6]

8. Cho list sau: lst\_data = [

[1,2,3],

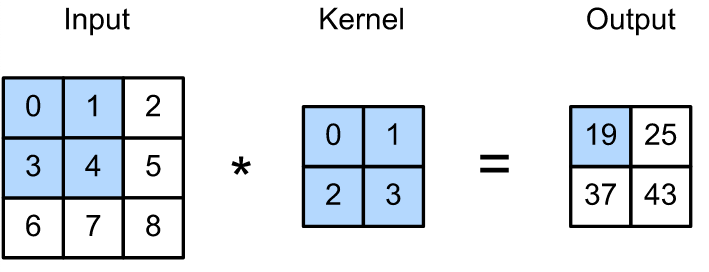
[2,3,5],

[1,2,1]

]

Hãy tìm số lớn nhất/bé nhất có trong lst\_data

9. Hãy cài đặt code python theo cách tính Convolution sau: (Được dùng numpy, for, if else, không dùng PyTorch)



Input = a.b, Kernal = m.n

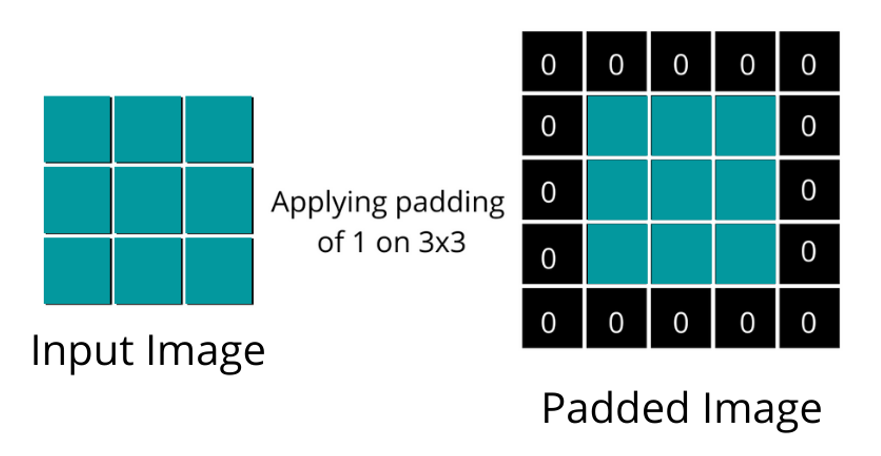
Output = (a-m+1).(b-n+1)

* Cách tính: áp kernel vào input, sau đó nhân tương ứng các giá trị hàng cột lấy tổng sẽ ra kết quả của output ở cột 1 hàng 1, sau đó trượt qua ô kế tiếp thực hiện tương tự.
* Ví dụ:

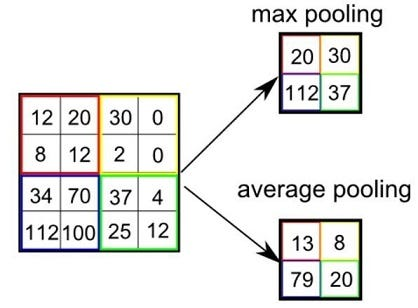
(\*) Output hàng 1 cột 1: 0\*0 + 1\*1 + 3\*2 + 4\*3 = 19

Output hàng 1 cột 2: 1\*0 + 2\*1 + 4\*2 + 5\* 3 = 25

Tương tự với các vị trí kế tiếp

10. Hãy thực hiện Padding cho input ở câu 9

11. Dùng kết quả padding ở câu 10, thực hiện nhân convolution kèm áp dùng Max Pooling hoặc Avg Pooling



12. Hãy cài đặt cách tính số euler ( số e) bằng phương pháp xấp xỉ Taylor.