

# AI VIET NAM - AIO 2023

## EXERCISE: BASIC PYTHON



**AI VIET NAM**  
@aivietnam.edu.vn



- **Python print() function**
  - Giới thiệu hàm print() cơ bản
- **Exercise1: Viết function tìm thể tích**
  - Hình lập phương
  - Hình hộp chữ nhật
  - Hình trụ tròn
  - Hình cầu
- **Exercise2: Viết function đánh giá classification model bằng f1-score**
- **Exercise3: Viết function tính kết quả theo các activation function**
  - Binary Step Function
  - Sigmoid Function
  - Elu Function



- **Exercise4: Viết function tính loss theo regression loss function**
  - Mean Absolute Error (MAE)
  - Mean Square Error (MSE)
  - Root Mean Square Error (RMSE)
- **Optional Exercises: Giới thiệu idea các bài tập thêm**
  - Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence
  - Exerices6: Viết function xấp xỉ các hàm lượng giác
  - Exercise7: Viết function đảo ngược các vị trí 1 số nguyên dương cho trước



- **Python print() function**
  - Giới thiệu hàm print() cơ bản
- **Exercise1: Viết function tìm thể tích**
  - Hình lập phương
  - Hình hộp chữ nhật
  - Hình trụ tròn
  - Hình cầu
- **Exercise2: Viết function đánh giá classification model bằng f1-score**
- **Exercise3: Viết function tính kết quả theo các activation function**
  - Binary Step Function
  - Sigmoid Function
  - Elu Function



# Python print() function



```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

- **objects:** Các objects muốn in ra. \* thể hiện có thể in được nhiều hơn 1 object
- **sep (optional):** Khi in nhiều object, các ký tự mong muốn dùng để phân chia các objects. Default là ' '
- **end (optional):** Các ký tự muốn in ở cuối cùng (cuối line). Default là '\n'
- **file (optional):** Phải là object với write method. Default là sys.stdout
- **flush (optional):** nếu True output là flushed hoặc buffered nếu False
- **Note:** Các arguments **sep, end, file, flush** là keyword arguments
- **Note:** Các objects sẽ được convert sang string type

# Python print() function



```
99 # print 1 object
100 print("Xin Chào AI VN")
101
102 b = 5
103 # print 2 object
104 print("a =", b)
105
106 # print 4 object
107 c = 999.9
108 print('a =', b, "c =", c)
```

```
Xin Chào AI VN
a = 5
a = 5 c = 999.9
```

- sep = ' ': Giữa các object có một space
- end = '\n' : Xuống hàng ở mỗi phần cuối của object cuối cùng
- file = sys.stdout : output được print trên screen
- flush The stream is not forcibly flushed.

```
127 sourceFile = open('python.txt', 'w')
128 print('Content in txt file', file = sourceFile)
129 sourceFile.close()
```

```
113 a = 'home'
114 b = 'Khoa'
115 c = 'AI_2022'
116 d = 'Exercisel'
117 print("path = ", a, b, c, d, sep='/')
```

```
path = /home/Khoa/AI_2022/Exercisel
```

- sep = '/' : Giữa các object có một ký tự '/'
- end = '\n' : Xuống hàng ở mỗi phần cuối của object cuối cùng

```
121 print("line1", end='- ')
122 print("line2", end='- ')
123 print("line3")
```

```
line1-line2-line3
```

- sep = ' ': Giữa các object có một space
- end = '- ' : Thêm ký tự '-' vào phần cuối cùng của object cuối cùng

# Python print() function



- **f-strings in python**

```
131 var_name = 'vname'
132 f"This is an f-string {var_name} and {var_name}."
```

This is an f-string vname and vname.

Đặt biến vào trong giữa {}. Tại runtime tất cả các biến này sẽ được thay bằng giá trị tương ứng

```
135 num1 = 4
136 num2 = 5
137 print(f"The product of {num1} and {num2} is {num1 * num2}.")
```

The product of 4 and 5 is 20.

Có thể là variable hoặc expression

```
142 def mul(num1, num2):
143     return num1 * num2
144 num1 = 4
145 num2 = 5
146 print(f"The product of {num1} and {num2} is {mul(num1, num2)}.")
```

The product of 4 and 5 is 20.

Có thể là variable hoặc call function





- **Python print() function**
  - Giới thiệu hàm print() cơ bản
- **Exercise1: Viết function tìm thể tích**
  - Hình lập phương
  - Hình hộp chữ nhật
  - Hình trụ tròn
  - Hình cầu
- **Exercise2: Viết function đánh giá classification model bằng f1-score**
- **Exercise3: Viết function tính kết quả theo các activation function**
  - Binary Step Function
  - Sigmoid Function
  - Elu Function





# Exercise1: Viết function tìm thể tích



**Viết 4 functions thực hiện tính thể tích của các khối sau:**

- Hình lập phương  $V = s^3$  (s là độ dài của cạnh)
  - input function: s
  - output function: V
- Hình hộp chữ nhật  $V = l * w * h$  (l là chiều dài, w là chiều rộng, h: là chiều cao)
  - input function: l, w, h
  - output function: V
- Hình trụ tròn:  $V = \pi * r^2 * h$  (r là bán kính đáy, h là chiều cao)
  - input function: r, h
  - output function: V
- Hình cầu:  $V = \frac{4}{3} * \pi * r^3$  (r bán kính hình cầu)
  - input function: r
  - output function: V

NOTE: Tùy thuộc vào giá trị  $\pi$  các bạn định nghĩa mà kết quả có thể sai lệch. Ví dụ tự định nghĩa  $\pi = 3.14$  hoặc dùng math.pi  $\pi = 3.141592653589793$

```
1 # Examples
2 calc_cube_volume(s=3)
3 >> 27
4
5 calc_rectangular_prism_volume(l=1, w=2, h=3)
6 >> 6
7
8 calc_cylinder_volume(r=1.2, h=4)
9 >> 18.0864
```

# Exercise1: Viết function tìm thể tích



- **Hiểu yêu cầu đề bài**

Viết 4 Functions

Tính Thể Tích

Hình Cầu  
Hình Trụ Tròn  
Hình Lập Phương  
Hình Hộp Chữ Nhật

**Thông tin thêm:**  
**- PI tùy ý định nghĩa**

Viết 4 functions thực hiện tính thể tích của các khối sau:

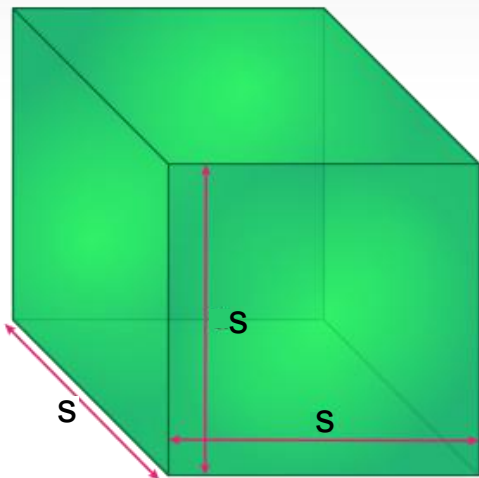
- Hình lập phương  $V = s^3$  (s là độ dài của cạnh)
  - input function: s
  - output function: V
- Hình hộp chữ nhật  $V = l * w * h$  (l là chiều dài, w là chiều rộng, h: là chiều cao)
  - input function: l, w, h
  - output function: V
- Hình trụ tròn:  $V = \pi * r^2 * h$  (r là bán kính đáy, h là chiều cao)
  - input function: r, h
  - output function: V
- Hình cầu:  $V = \frac{4}{3} * \pi * r^3$  (r bán kính hình cầu)
  - input function: r
  - output function: V

NOTE: Tùy thuộc vào giá trị  $\pi$  các bạn định nghĩa mà kết quả có thể sai lệch. Ví dụ tự định nghĩa  $\pi = 3.14$  hoặc dùng `math.pi`  $\pi = 3.141592653589793$

# Exercise1: Viết function tìm thể tích

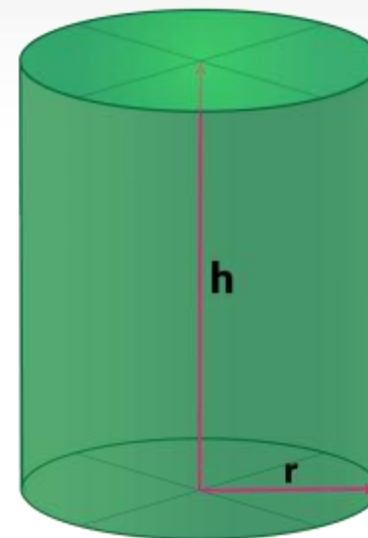


## • Xác Định Input và Output



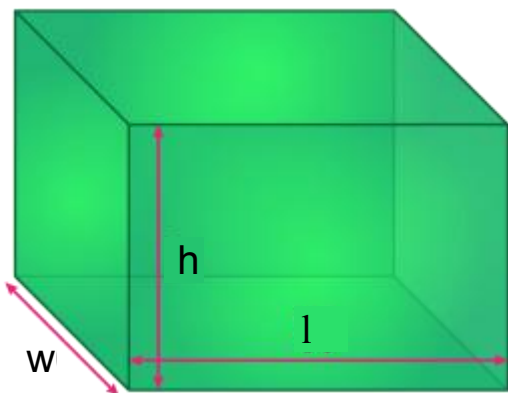
$$V = s^3$$

- Input: nhận s
- Output:  $V = s^3$



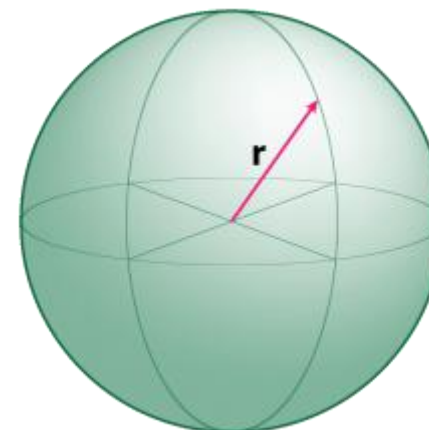
$$V = \pi r^2 h$$

- Input: nhận r và h
- Output:  $V = \pi r^2 h$
- Cần xác định  $\pi$



$$V = l * w * h$$

- Input: nhận l, w, h
- Output:  $V = l * w * h$



$$V = \frac{4}{3} \pi r^3$$

- Input: nhận r và h
- Output:  $V = \frac{4}{3} \pi r^3$
- Cần xác định  $\pi$

AI

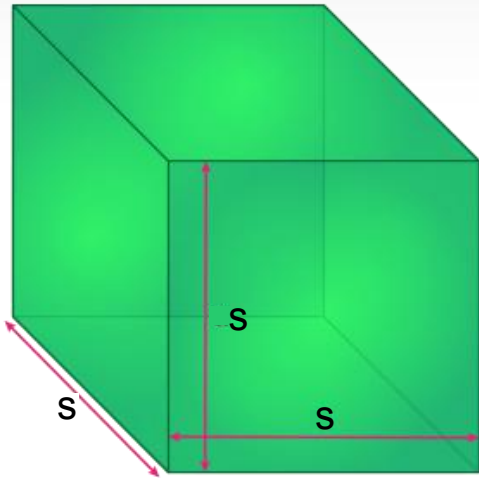
AI VIET NAM

@aivietnam.edu.vn

# Exercise1: Viết function tìm thể tích

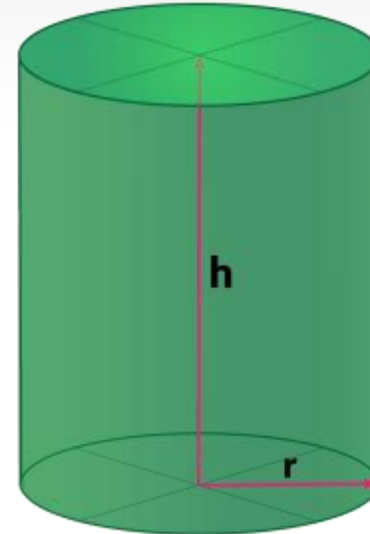


- Pseudocode**



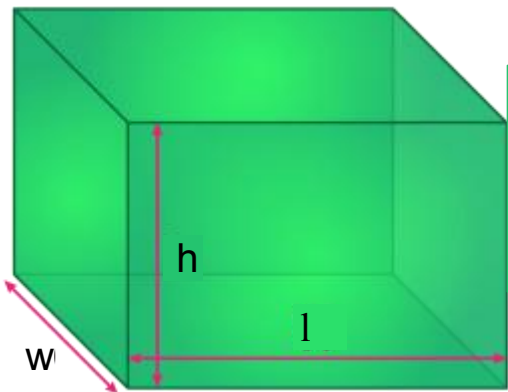
$$V = s^3$$

```
FUNCTION calc_cube_volume(s)  
    V = s*s*s  
RETURN V
```



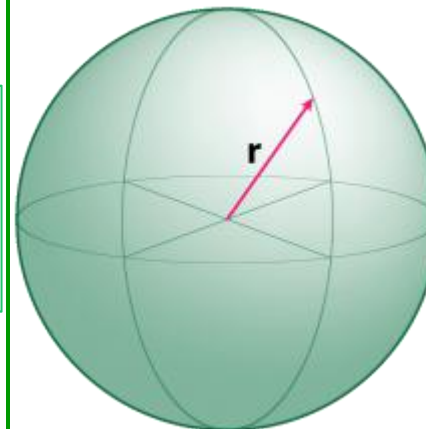
$$V = \pi r^2 h$$

```
FUNCTION calc_cylinder_volume(r,h)  
    pi = 3.14  
    V = pi*r*r*h  
RETURN V
```



$$V = l * w * h$$

```
FUNCTION  
calc_rectangular_prism_volume(l,w,h)  
    V = l*w*h  
RETURN V
```



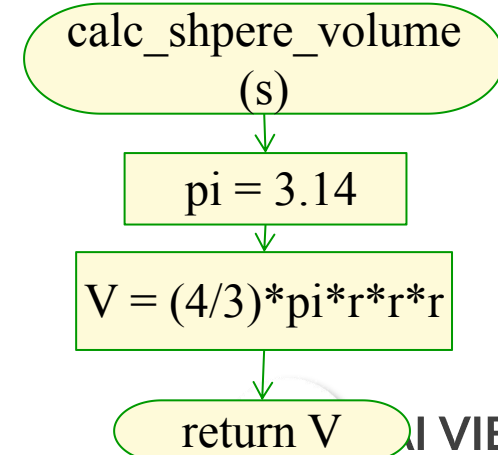
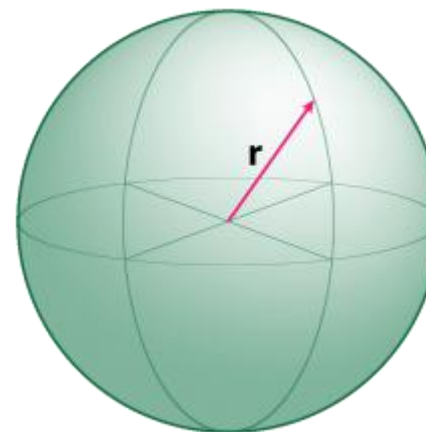
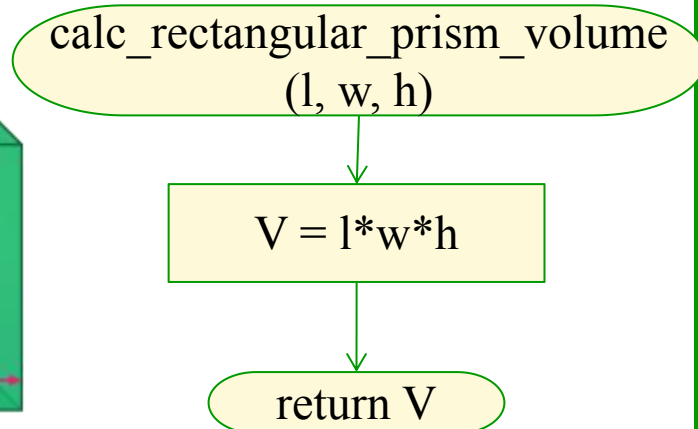
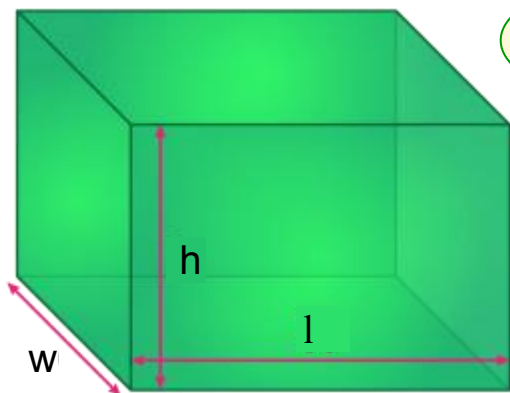
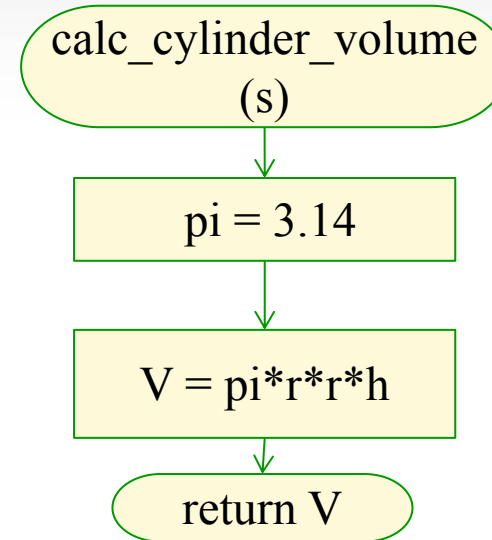
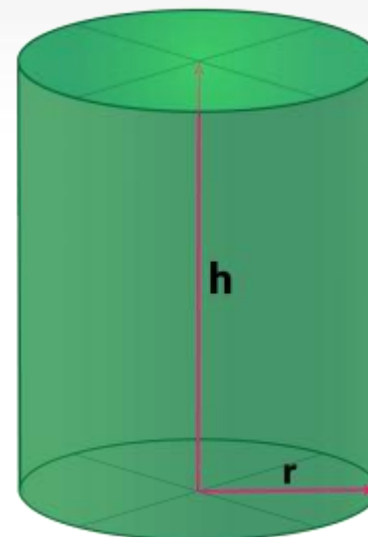
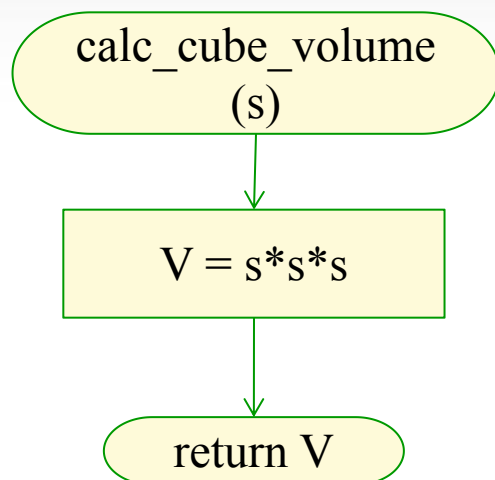
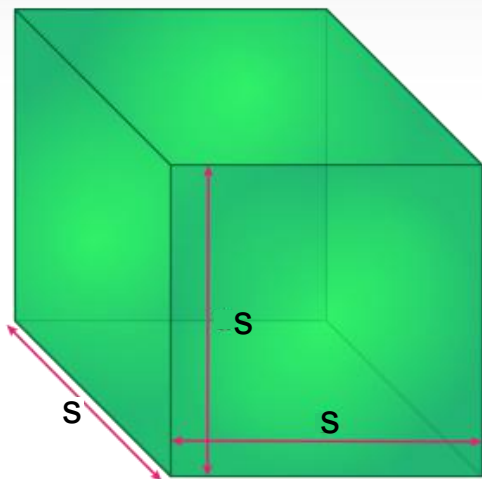
$$V = \frac{4}{3} \pi r^3$$

```
FUNCTION calc_shpere_volume(r)  
    pi = 3.14  
    V = (4/3)*pi*r*r*r  
RETURN V
```

# Exercise1: Viết function tìm thể tích



- Flowchart





- **Python print() function**
  - Giới thiệu hàm print() cơ bản
- **Exercise1: Viết function tìm thể tích**
  - Hình lập phương
  - Hình hộp chữ nhật
  - Hình trụ tròn
  - Hình cầu
- **Exercise2: Viết function đánh giá classification model bằng f1-score**
- **Exercise3: Viết function tính kết quả theo các activation function**
  - Binary Step Function
  - Sigmoid Function
  - Elu Function



# Exercise2: Viết function đánh giá classification model bằng f1-score



## 2. Viết function thực hiện đánh giá classification model bằng F1-Score.

- Precision =  $\frac{TP}{TP + FP}$
- Recall =  $\frac{TP}{TP + FN}$
- F1-score =  $2 * \frac{Precision * Recall}{Precision + Recall}$
- Input: function nhận 3 giá trị **tp**, **fp**, **fn**
- Output: print ra kết quả của **Precision**, **Recall**, và **F1-score**

### NOTE: Đề bài yêu cầu các điều kiện sau

- Phải kiểm tra giá trị nhận vào **tp**, **fp**, **fn** là type **int**, nếu là type khác thì print ví dụ check **fn** là float, print '**fn must be int**' và thoát hàm hoặc dừng chương trình.
- Yêu cầu **tp**, **fp**, **fn** phải đều lớn hơn 0, nếu không thì print '**tp and fp and fn must be greater than zero**' và thoát hàm hoặc dừng chương trình

```
1 # Examples
2 calc_f1_score(tp=2, fp=3, fn=4)
3 >> precision is 0.4
4 recall is 0.3333333333333333
5 f1-score is 0.3636363636363636
6
7 calc_f1_score(tp='a', fp=3, fn=4)
8 >> tp must be int
9
10
11 calc_f1_score(tp=2, fp='a', fn=4)
12 >> fp must be int
13
14
15 calc_f1_score(tp=2, fp=3, fn='a')
16 >> fn must be int
17
18 calc_f1_score(tp=2, fp=3, fn=0)
19 >> tp and fp and fn must be greater than zero
20
21 calc_f1_score(tp=2.1, fp=3, fn=0)
22 >> tp must be int
```



## Exercise2: Viết function đánh giá calssification model bằng f1-score



### 2. Viết function thực hiện đánh giá classification model bằng F1-Score.

$$\bullet \text{ Precision} = \frac{TP}{TP+FP} = \frac{TP}{\text{all detections}}$$

$$\bullet \text{ Recall} = \frac{TP}{TP+FN} = \frac{TP}{\text{all ground truths}}$$

$$\bullet \text{ F1 - score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Input: function nhận 3 giá trị **tp, fp, fn**
- Output: print ra kết quả của **Precision, Recall, và F1-score**

### NOTE: Đề bài yêu cầu các điều kiện sau

- Phải kiểm tra giá trị nhận vào **tp, fp, fn** là type **int**, nếu là type khác thì print ví dụ check fn là float, print '**fn must be int**' và thoát hàm hoặc dừng chương trình.
- Yêu cầu **tp, fp, fn** phải đều lớn hơn 0, nếu không thì print '**tp and fp and fn must be greater than zero**' và thoát hàm hoặc dừng chương trình

```
1 # Examples
2 calc_f1_score(tp=2, fp=3, fn=4)
3 >> precision is 0.4
4 recall is 0.3333333333333333
5 f1-score is 0.3636363636363636
6
7 calc_f1_score(tp='a', fp=3, fn=4)
8 >> tp must be int
9
10
11 calc_f1_score(tp=2, fp='a', fn=4)
12 >> fp must be int
13
14
15 calc_f1_score(tp=2, fp=3, fn='a')
16 >> tp must be int
17
18 calc_f1_score(tp=2, fp=3, fn=0)
19 >> tp and fp and fn must be greater than zero
20
21 calc_f1_score(tp=2.1, fp=3, fn=0)
22 >> tp must be int
```

## Exercise2: Viết function đánh giá classification model bằng f1-score



- **Hiểu yêu cầu đề bài**

Viết 1 Functions

Tính F1-Score

Hàm nhận 3 parameters  
tp, fp, và fn

Print kết quả  
Precision, Recall, F1-score

**Thông tin thêm:**

- Các ràng buộc và xử lý lỗi
- tp, fp, fn thuộc type int
- tp, fp, fn > 0

2. **Viết function** thực hiện **đánh giá** classification model **bằng F1-Score.**

- $Precision = \frac{TP}{TP+FP} = \frac{TP}{all\ detections}$

- $Recall = \frac{TP}{TP+FN} = \frac{TP}{all\ ground\ truths}$

- $F1 - score = \frac{2*Precision*Recall}{Precision+Recall}$

- Input: function nhận 3 giá trị **tp, fp, fn**
- Output: print ra kết quả của **Precision, Recall, và F1-score**

**NOTE:** Đề bài yêu cầu các điều kiện sau

- Phải **kiểm tra** giá trị nhận vào **tp, fp, fn** là **type int**, nếu là type khác thì print ví dụ check fn là float, print '**fn must be int**' và thoát hàm hoặc dừng chương trình.
- Yêu cầu **tp, fp, fn phải đều lớn hơn 0**, nếu không thì print '**tp and fp and fn must be greater than zero**' và thoát hàm hoặc dừng chương trình

## Exercise2: Viết function đánh giá classification model bằng f1-score

### • Pseudocode

$$\begin{aligned} \bullet \text{ Precision} &= \frac{TP}{TP+FP} = \frac{TP}{\text{all detections}} \\ \bullet \text{ Recall} &= \frac{TP}{TP+FN} = \frac{TP}{\text{all ground truths}} \\ \bullet \text{ F1 - score} &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

**NOTE:** Đề bài yêu cầu các điều kiện sau

- Phải kiểm tra giá trị nhận vào **tp, fp, fn** là **type int**, nếu là type khác thì print ví dụ check fn là float, print '**fn must be int**' và thoát hàm hoặc dừng chương trình.
- Yêu cầu **tp, fp, fn phải đều lớn hơn 0**, nếu không thì print '**tp and fp and fn must be greater than zero**' và thoát hàm hoặc dừng chương trình

```
FUNCTION calc_f1_score(tp, fp, fn)
    precision = tp/(tp+fp)
    recall = tp/(tp+fn)
    f1_score = 2*(precision*recall)/(precision+recall)

    PRINT precision
    PRINT recall
    PRINT f1_score

ENDFUNCTION
```

```
FUNCTION calc_f1_score(tp, fp, fn)
    IF type of tp is not int type THEN
        PRINT "tp must be int"
        RETURN
    IF type of fp is not int type THEN
        PRINT "fp must be int"
        RETURN
    IF type of fn is not int type THEN
        PRINT "fn must be int"
        RETURN
    IF not (tp > 0) and not (fp > 0) and not (fn > 0) THEN
        PRINT "tp and fp and fn must be greater than zero"
        RETURN
    ENDIF

    precision = tp/(tp+fp)
    recall = tp/(tp+fn)
    f1_score = 2*(precision*recall)/(precision+recall)

    PRINT precision
    PRINT recall
    PRINT f1_score

ENDFUNCTION
```

## Exercise2: Viết function đánh giá calssification model bằng f1-score



```
FUNCTION calc_f1_score(tp, fp, fn)
```

```
  IF type of tp is not int type THEN
```

```
    PRINT "tp must be int"
```

```
    RETURN
```

```
  IF type of fp is not int type THEN
```

```
    PRINT "fp must be int"
```

```
    RETURN
```

```
  IF type of fn is not int type THEN
```

```
    PRINT "fn must be int"
```

```
    RETURN
```

```
  IF not (tp > 0) and not (fp > 0) and not (fn > 0) THEN
```

```
    PRINT "tp and fp and fn must be greater than zero"
```

```
    RETURN
```

```
  ENDIF
```

```
  precision = tp/(tp+fp)
```

```
  recall = tp/(tp+fn)
```

```
  f1_score = 2*(precision*recall)/(precision+recall)
```

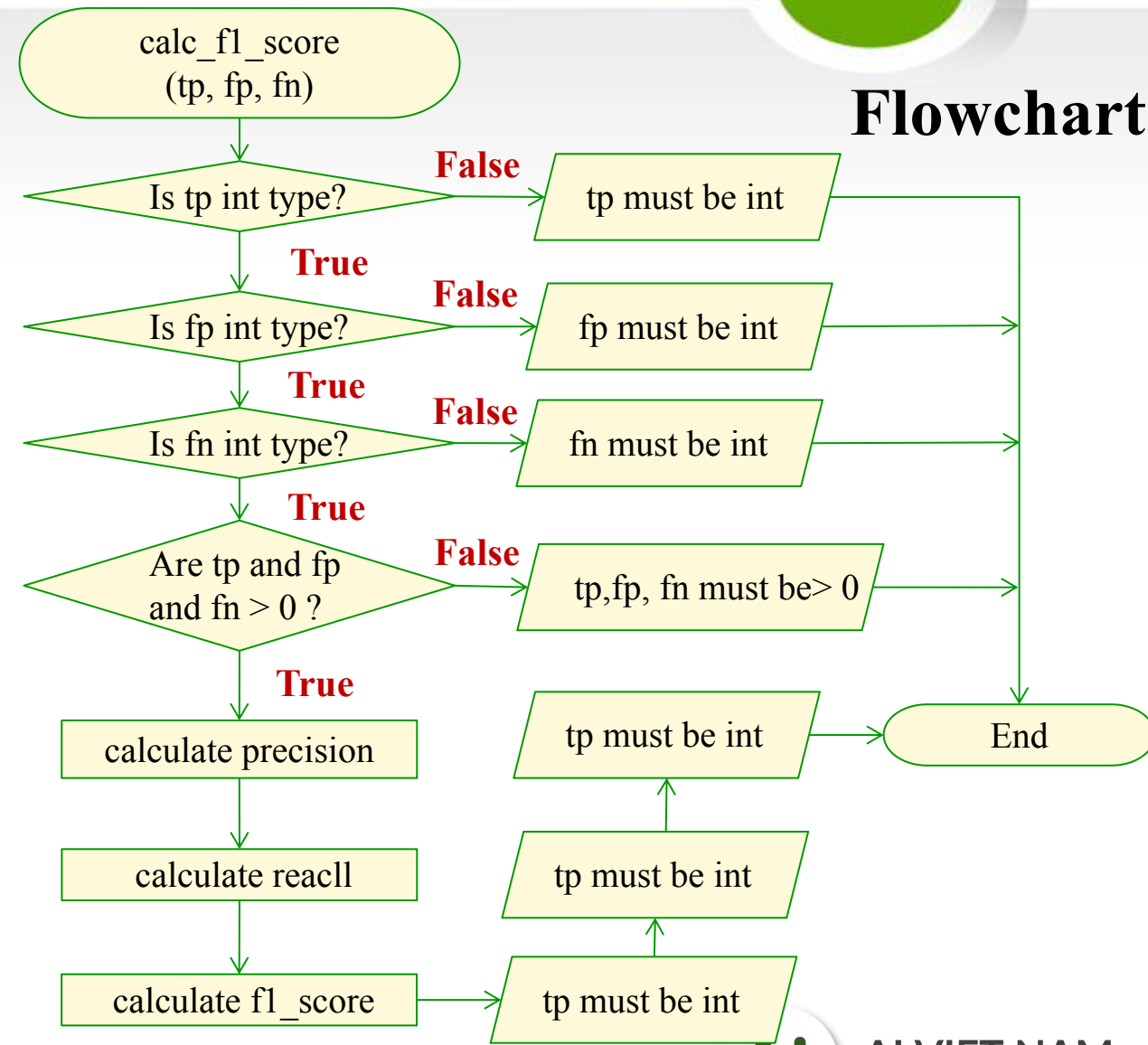
```
  PRINT precision
```

```
  PRINT recall
```

```
  PRINT f1_score
```

```
ENDFUNCTION
```

### Flowchart





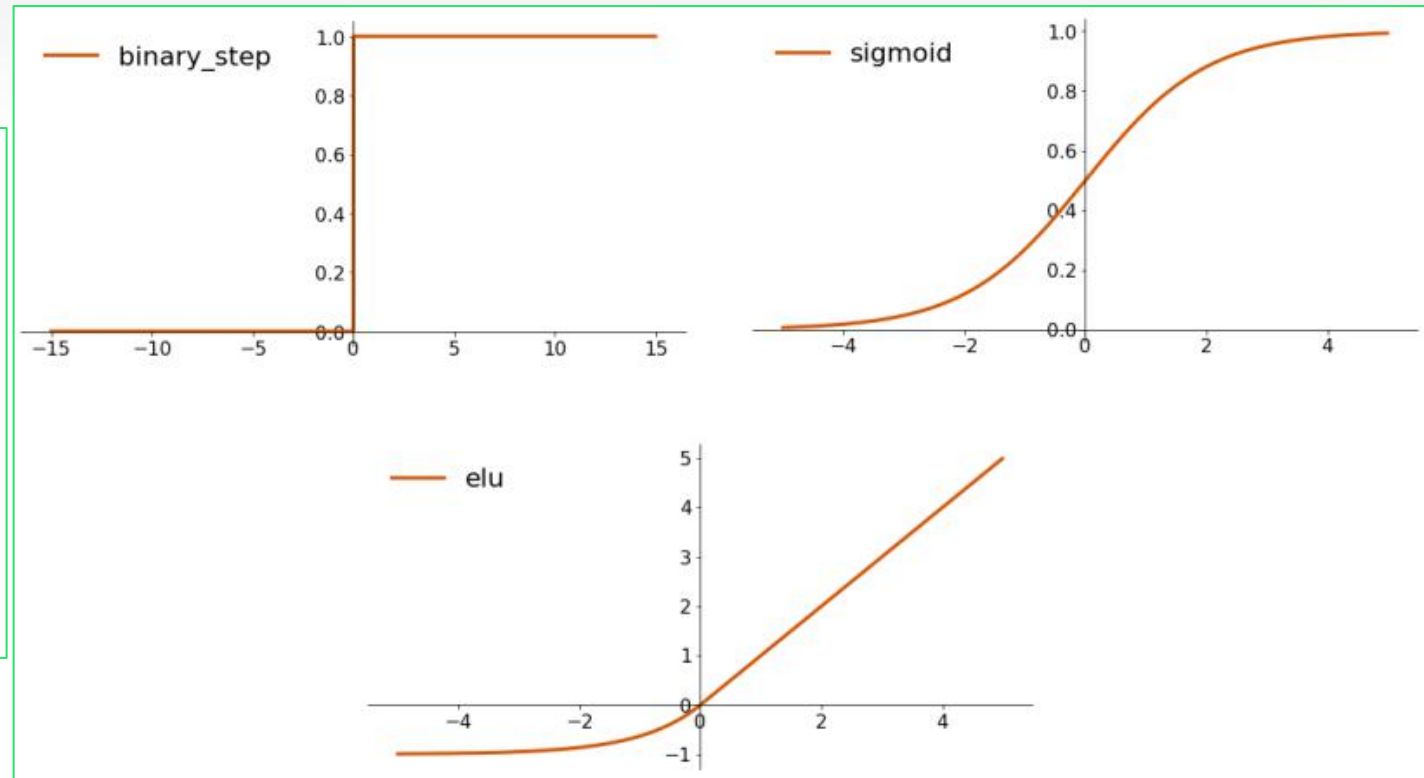
- **Python print() function**
  - Giới thiệu hàm print() cơ bản
- **Exercise1: Viết function tìm thể tích**
  - Hình lập phương
  - Hình hộp chữ nhật
  - Hình trụ tròn
  - Hình cầu
- **Exercise2: Viết function đánh giá classification model bằng f1-score**
- **Exercise3: Viết function tính kết quả theo các activation function**
  - Binary Step Function
  - Sigmoid Function
  - Elu Function

# Exercise3: Viết function tính kết quả theo các activation function



## 3. Viết function mô phỏng theo 3 activation function:

- Binary Step Function  $f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
- Sigmoid Function  $f(x) = \frac{1}{1+e^{-x}}$
- Elu Function  $f(x) = \begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$





# Exercise3: Viết function tính kết quả theo các activation function



- Input:
  - Người dùng nhập giá trị  $x$
  - Người dùng nhập tên **activation function** chỉ có 3 loại (binary, sigmoid, elu)
- Output: Kết quả  $f(x)$  (x khi đi qua activation function tương ứng theo activation function name). Ví dụ **nhập  $x=3$ , activation\_function = 'binary'. Output: print 'binary: f(3)=1'**

**NOTE:** Lưu ý các điều kiện sau:

- Dùng function `is_number` được cung cấp sẵn để kiểm tra  $x$  có hợp lệ hay không (vd:  $x='10'$ , `is_number(x)` sẽ trả về True ngược lại là False). Nếu không hợp lệ print ' $x$  must be a number' và dùng chương trình.

- Kiểm tra **activation function name** có hợp lệ hay không nằm trong 3 tên (binary, sigmoid, elu). Nếu không print '`ten_function_user is not supported`' (vd người dùng nhập 'relu' thì print 'relu is not supported')

- Convert  $x$  sang **float** type
- Thực hiện theo công thức với activation name tương ứng. Print ra kết quả
- Dùng `math.e` để lấy số  $e$
- $\alpha = 0.01$

```
1 # Given
2 def is_number(n):
3     try:
4         float(n)      # Type-casting the string to 'float'.
5                       # If string is not a valid 'float',
6                       # it'll raise 'ValueError' exception
7     except ValueError:
8         return False
9     return True
```

Code Listing 3: Cho trước hàm `is_number`

```
1 exercise3()
2 >> Input x = 1.5
3 Input activation Function (binary|sigmoid|elu): sigmoid
4 sigmoid: f(1.5) = 0.8175744761936437
5
6 exercise3()
7 >> Input x = abc
8 x must be a number
9
10 exercise3()
11 >> Input x = 1.5
12 Input activation Function (binary|sigmoid|elu): relu
13 relu is not supported
```



# Exercise3: Viết function tính kết quả theo các activation function

## • Hiểu yêu cầu đề bài

Viết 1 Functions

Tính 1 trong 3 activation function từ yêu cầu từ user

Hàm nhận 2 input từ user x và tên function

Print tên function và kết quả với x

Thông tin thêm:

- Chỉ nhận x là con số
- Chỉ nhận 1 trong 3 tên activation function mà hàm hỗ trợ
- x khi tính toán nên ở dạng float
- Dùng math module để lấy số e
- Dùng  $\alpha = 0.01$

• Binary Step Function  $f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$

• Sigmoid Function  $f(x) = \frac{1}{1+e^{-x}}$

• Elu Function  $f(x) = \begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$

• Input:

- Người dùng nhập giá trị x
- Người dùng nhập tên activation function chỉ có 3 loại (binary, sigmoid, elu)

• Output: Kết quả  $f(x)$  (x khi đi qua activation function tương ứng theo activation function name). Ví dụ nhập x=3, activation\_function = 'binary'. Output: print 'binary: f(3)=1'

- Convert x sang float type
- Dùng math.e để lấy số e
- $\alpha = 0.01$

• Dùng function is\_number được cung cấp sẵn để kiểm tra x có hợp lệ hay không (vd: x='10', is\_number(x) sẽ trả về True ngược lại là False). Nếu không hợp lệ print 'x must be a number' và dừng chương trình.

• Kiểm tra activation function name có hợp lệ hay không nằm trong 3 tên (binary, sigmoid, elu). Nếu không print 'ten\_function\_user is not supported' (vd người dùng nhập 'relu' thì print 'relu is not supported')

```

FUNCTION calc_activation_func(x, act_name)
    result = None
    IF act_name == 'binary' THEN
        result = calc_bin(x)
    ELSE IF act_name == 'sigmoid' THEN
        result = calc_sig(x)
    ELSE IF act_name == 'elu' THEN
        result = calc_elu(x)
    RETURN result
ENDFUNCTION

```

```

FUNCTION calc_sig(x)
    result = 1./(1+math.e**(-x))
    RETURN result
ENDFUNCTION

```

```

FUNCTION calc_bin(x)
    IF x < 0 THEN
        result = 0.
    ELSE
        result = 1.
    ENDIF
    RETURN result
ENDFUNCTION

```

```

FUNCTION calc_elu(x)
    alpha = 0.01
    IF x < 0 THEN
        result = alpha*(math.e**x - 1)
    ELSE
        result = x
    ENDIF
    RETURN result
ENDFUNCTION

```

```

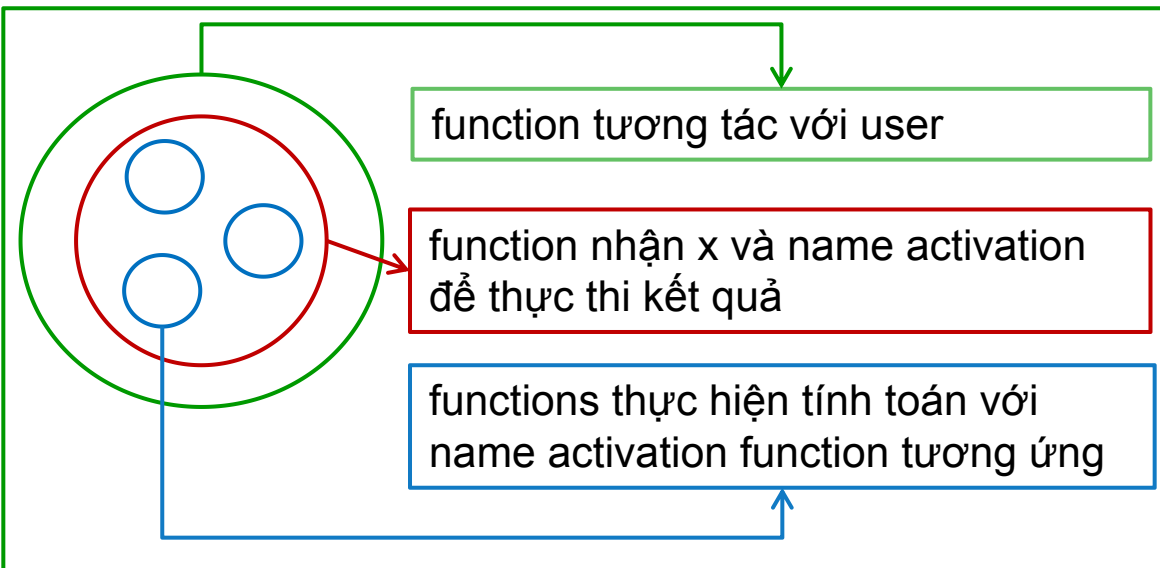
FUNCTION exercise3 ()
    x = input('Input x = ')
    IF not is_number(x) THEN
        PRINT "x must be a number"
        RETURN
    ENDIF

    act_name = input('activation function name: ')
    cast x to float
    result = calc_activation_func(x, act_name)

    IF act_name is valid (result is None) THEN
        PRINT act_name + "is not supported"
    ELSE
        PRINT act_name + string(x) + string(result)
    ENDIF

ENDFUNCTION

```



# Exercise3: Viết function tính kết quả theo các activation function



## Flowchart

```
FUNCTION exercise3 ()
```

```
  x = input('Input x = ')
```

```
  IF not is_number(x) THEN
```

```
    PRINT "x must be a number"
```

```
    RETURN
```

```
  ENDIF
```

```
  act_name = input('activation function name: ')
```

```
  cast x to float
```

```
  result = calc_activation_func(x, act_name)
```

```
  IF act_name is valid (result is None) THEN
```

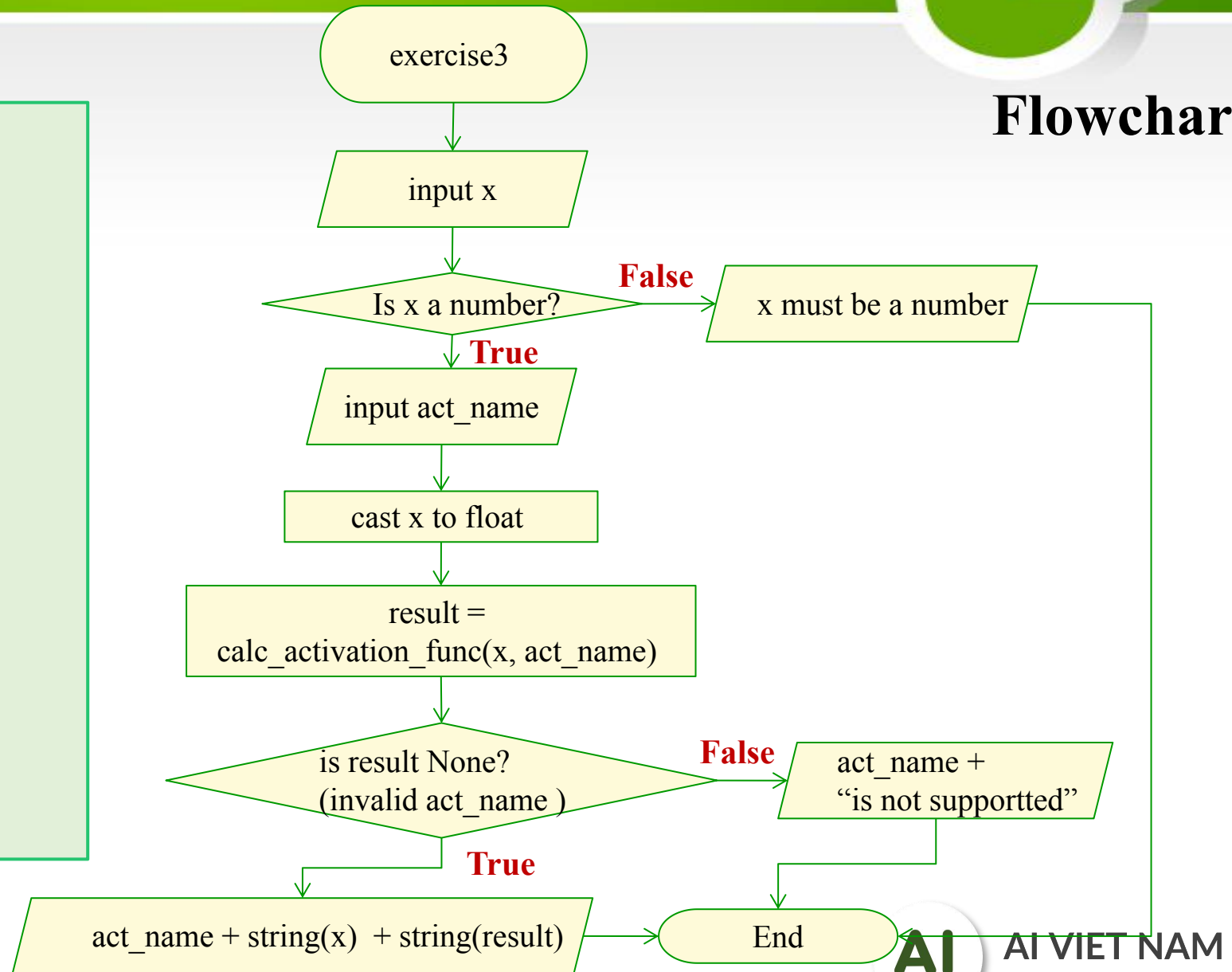
```
    PRINT act_name + "is not supported"
```

```
  ELSE
```

```
    PRINT act_name + string(x) + string(result)
```

```
  ENDIF
```

```
ENDFUNCTION
```





- **Exercise4: Viết function tính loss theo regression loss function**
  - Mean Absolute Error (MAE)
  - Mean Square Error (MSE)
  - Root Mean Square Error (RMSE)
- **Optional Exercises: Giới thiệu idea các bài tập thêm**
  - Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence
  - Exerices6: Viết function xấp xỉ các hàm lượng giác
  - Exercise7: Viết function đảo ngược các vị trí 1 số nguyên dương cho trước

# Exercise4: Viết function tính loss theo regression loss function



## 4. Viết function lựa chọn regression loss function để tính loss :

- $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$

- $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$

- $RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$

- **n** chính là **số lượng samples (num\_samples)**, với **i** là mỗi sample cụ thể. Ở đây các bạn có thể hiểu là cứ mỗi **i** thì sẽ có **1 cặp**  $y_i$  là **target** và  $\hat{y}_i$  là **predict**.

- **Input:**

- Người dùng **nhập số lượng sample (num\_samples)** được tạo ra (**chỉ nhận integer numbers**)
- Người dùng **nhập loss name (MAE, MSE, RMSE-(optional))** chỉ cần **MAE và MSE**, bạn nào muốn làm thêm RMSE đều được.

- **Output:** Print ra **loss name, sample, predict, target, loss**

- **loss name:** là loss mà người dùng chọn

- **sample:** là thứ tự sample được tạo ra (ví dụ num\_samples=5, thì sẽ có 5 samples và mỗi sample là sample-0, sample-1, sample-2, sample-3, sample-4)

- **predict:** là số mà model dự đoán (chỉ cần dùng random tạo random một số trong range [0,10))

- **target:** là số target mà mong muốn model dự đoán đúng (chỉ cần dùng random tạo random một số trong range [0,10))

- **loss:** là kết quả khi đưa predict và target vào hàm loss

- **note:** ví dụ num\_sample=5 thì sẽ có 5 cặp predict và target

# Exercise4: Viết function tính loss theo regression loss function



- Dùng `.isnumeric()` method để kiểm tra `num_samples` có hợp lệ hay không (vd: `x='10'`, `num_samples.isnumeric()` sẽ trả về True ngược lại là False). **Không hợp lệ print 'number of samples must be an integer number'** và dùng chương trình.
- Dùng vòng lặp `for`, lặp lại `num_samples` lần. Mỗi lần dùng `random` modules tạo một con số ngẫu nhiên trong range `[0.0, 10.0)` cho `predict` và `target`. Sau đó đưa `predict` và `target` vào loss function và print ra kết quả mỗi lần lặp.
- Dùng `random.uniform(0,10)` để tạo ra một số ngẫu nhiên trong range `[0,10)`
- Giả xử người dùng luôn nhập đúng loss name **MSE, MAE, và RMSE** (đơn giản bước này để các bạn không cần check tên hợp lệ)
- Dùng `abs()` để tính trị tuyệt đối ví dụ `abs(-3)` sẽ trả về 3
- Dùng `math.sqrt()` để tính căn bậc 2

```
1 exercise4()
2 >> Input number of samples (integer number) which are generated: 5
3 Input loss name: RMSE
4 loss name: RMSE, sample: 0, pred: 6.659262156575629, target: 4.5905830130732355,
  loss: 4.279433398761796
5 loss name: RMSE, sample: 1, pred: 4.592264312227207, target: 8.447168720237958,
  loss: 14.860287994900718
6 loss name: RMSE, sample: 2, pred: 8.701801828625959, target: 9.280646891626386,
  loss: 0.3350616069599687
7 loss name: RMSE, sample: 3, pred: 4.799972972282257, target: 9.877147335937869,
  loss: 25.777699518961764
8 loss name: RMSE, sample: 4, pred: 0.20159822778697878, target: 5.540221923628147,
  loss: 28.50090296579681
9 final RMSE: 3.8406610234536727
10
11
12 exercise4()
13 >> Input number of samples (integer number) which are generated: aa
14 number of samples must be an integer number
```



# Exercise4: Viết function tính loss theo regression loss function

## • Hiểu yêu cầu đề bài

Viết 1 Functions

Tính 1 trong 3 loss function từ yêu cầu từ user

Hàm nhận 2 input từ user num\_samples và tên function

Print loss name, sample, predict, target, loss

Thông tin thêm:

- Dùng `.isnumeric()` kiểm tra `num_samples`
- `predict` và `target` ( $\hat{y}$ ,  $y$ ) được tạo ra random trong mỗi lần lặp
- Random trong range [0,10)

$$\bullet \text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$\bullet \text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\bullet \text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

• `n` chính là số lượng samples (`num_samples`), với `i` là mỗi sample cụ thể. Ở đây các bạn có thể hiểu là cứ mỗi `i` thì sẽ có 1 cặp  $y_i$  là target và  $\hat{y}_i$  là predict.

Công thức tính theo series cần vòng lặp `n` lần

• Input:

- Người dùng nhập số lượng sample (`num_samples`) được tạo ra (chỉ nhận integer numbers)
- Người dùng nhập loss name (MAE, MSE, RMSE-(optional)) chỉ cần MAE và MSE, bạn nào muốn làm thêm RMSE đều được.

• Output: Print ra loss name, sample, predict, target, loss

- Dùng `.isnumeric()` method để kiểm tra `num_samples` có hợp lệ hay không (vd: `x='10'`, `num_samples.isnumeric()` sẽ trả về True ngược lại là False). Không hợp lệ print 'number of samples must be an integer number' và dừng chương trình.
- Dùng vòng lặp `for`, lặp lại `num_samples` lần. Mỗi lần dùng `random` modules tạo một con số ngẫu nhiên trong range [0.0, 10.0) cho `predict` và `target`. Sau đó đưa `predict` và `target` vào loss function và print ra kết quả mỗi lần lặp.
- Dùng `random.uniform(0,10)` để tạo ra một số ngẫu nhiên trong range [0,10)
- Giả sử người dùng luôn nhập đúng loss name MSE, MAE, và RMSE (đơn giản bước này để các bạn không cần check tên hợp lệ)



# Exercise4: Viết function tính loss theo regression loss function



$$\bullet \text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$\bullet \text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\bullet \text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

• **n** chính là **số lượng samples (num\_samples)**, với **i** là mỗi sample cụ thể. Ở đây các bạn có thể hiểu là cứ mỗi **i** thì sẽ có **1 cặp**  $y_i$  là **target** và  $\hat{y}_i$  là **predict**.

$$\sqrt{a} + \sqrt{b} \neq \sqrt{a + b}$$

```
1 exercise4()
2 >> Input number of samples (integer number) which are generated: 5
3 Input loss name: RMSE
4 loss name: RMSE, sample: 0, pred: 6.659262156575629, target: 4.5905830130732355,
  loss: 4.279433398761796
5 loss name: RMSE, sample: 1, pred: 4.592264312227207, target: 8.447168720237958,
  loss: 14.860287994900718
6 loss name: RMSE, sample: 2, pred: 8.701801828625959, target: 9.280646891626386,
  loss: 0.3350616069599687
7 loss name: RMSE, sample: 3, pred: 4.799972972282257, target: 9.877147335937869,
  loss: 25.777699518961764
8 loss name: RMSE, sample: 4, pred: 0.20159822778697878, target: 5.540221923628147,
  loss: 28.50090296579681
9 final RMSE: 3.8406610234536727
10
11
12 exercise4()
13 >> Input number of samples (integer number) which are generated: aa
14 number of samples must be an integer number
```

loss của  
Squared Error

Final loss của  
RMSE

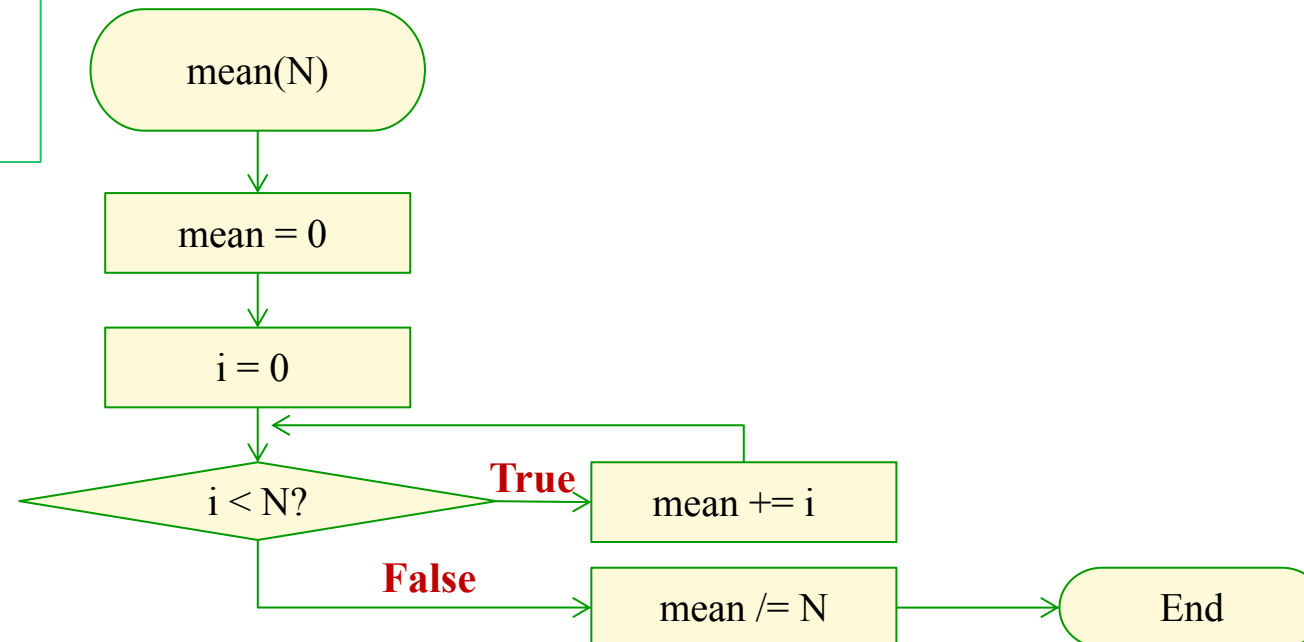
# Exercise4: Viết function tính loss theo regression loss function



- $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$
- $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
- $RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
- **n** chính là **số lượng samples (num\_samples)**, với **i** là mỗi sample cụ thể. Ở đây các bạn có thể hiểu là cứ mỗi **i** thì sẽ có **1 cặp**  $y_i$  là **target** và  $\hat{y}_i$  là **predict**.

$$mean = \frac{1}{N} \sum_{i=0}^{N-1} i, N > 0$$

```
FUNCTION mean(N)
  mean = 0
  FOR i start at 0 TO N - 1
    mean += i
  ENDFOR
  mean /= N
  RETURN mean
ENDFUNCTION
```



- **Pseudocode**

```
FUNCTION calc_ae(y, y_hat)
    result = absolute(y - y_hat)
    RETURN result
ENDFUNCTION
```

```
FUNCTION calc_se(y, y_hat)
    result = (y - y_hat)**2
    RETURN result
ENDFUNCTION
```

```
FUNCTION exercise4()
```

```
    num_samples = input('Input num_samples = ')
```

```
    IF not samples .isnumeric() THEN
```

```
        PRINT "number of samples must be an integer number"
```

```
        RETURN
```

```
    ENDIF
```

```
    loss_name = input('Input loss name: ')
```

```
    final_loss = 0
```

```
    cast num_samples to int
```

```
    FOR i start at 0 TO num_samples - 1
```

```
        pred_sample = random FROM 0 TO less than 10
```

```
        target_sample = random FROM 0 TO less than 10
```

```
        IF loss_name == 'MAE' THEN
```

```
            loss = calc_ae(sample , target_sample)
```

```
        ELSE IF loss_name == 'MSE' OR loss_name == 'RMSE' THEN
```

```
            loss = calc_se(sample , target_sample)
```

```
        ENDIF
```

```
        final_loss += loss
```

```
        PRINT loss_name + string(i) + string(pred_sample) + string(target_sample) + string(loss)
```

```
    ENDFOR
```

```
    final_loss /= num_samples
```

```
    IF loss_name == 'RMSE' THEN
```

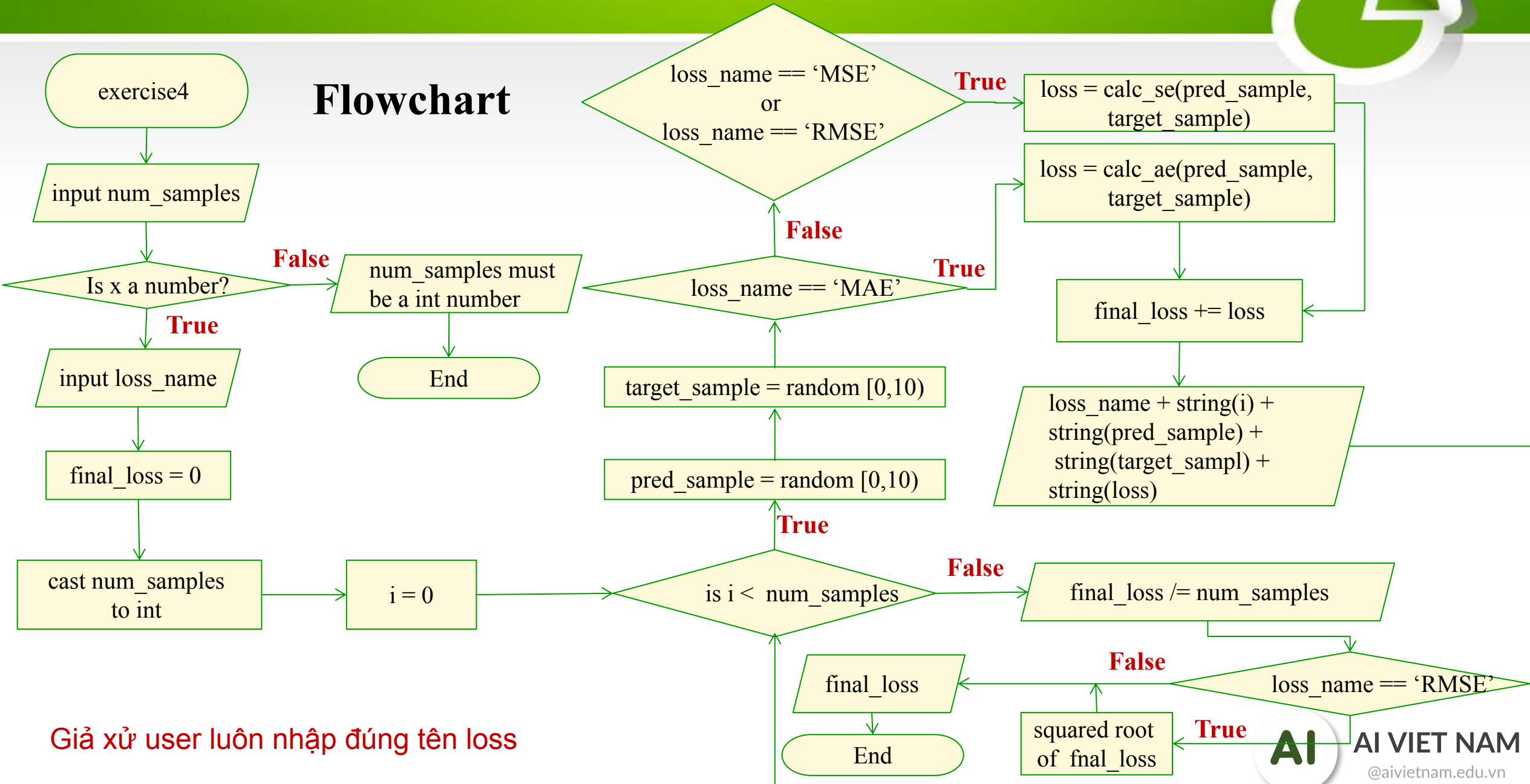
```
        final_loss = square root of final_loss
```

```
    ENDIF
```

```
    PRINT string(final_loss )
```

```
ENDFUNCTION
```

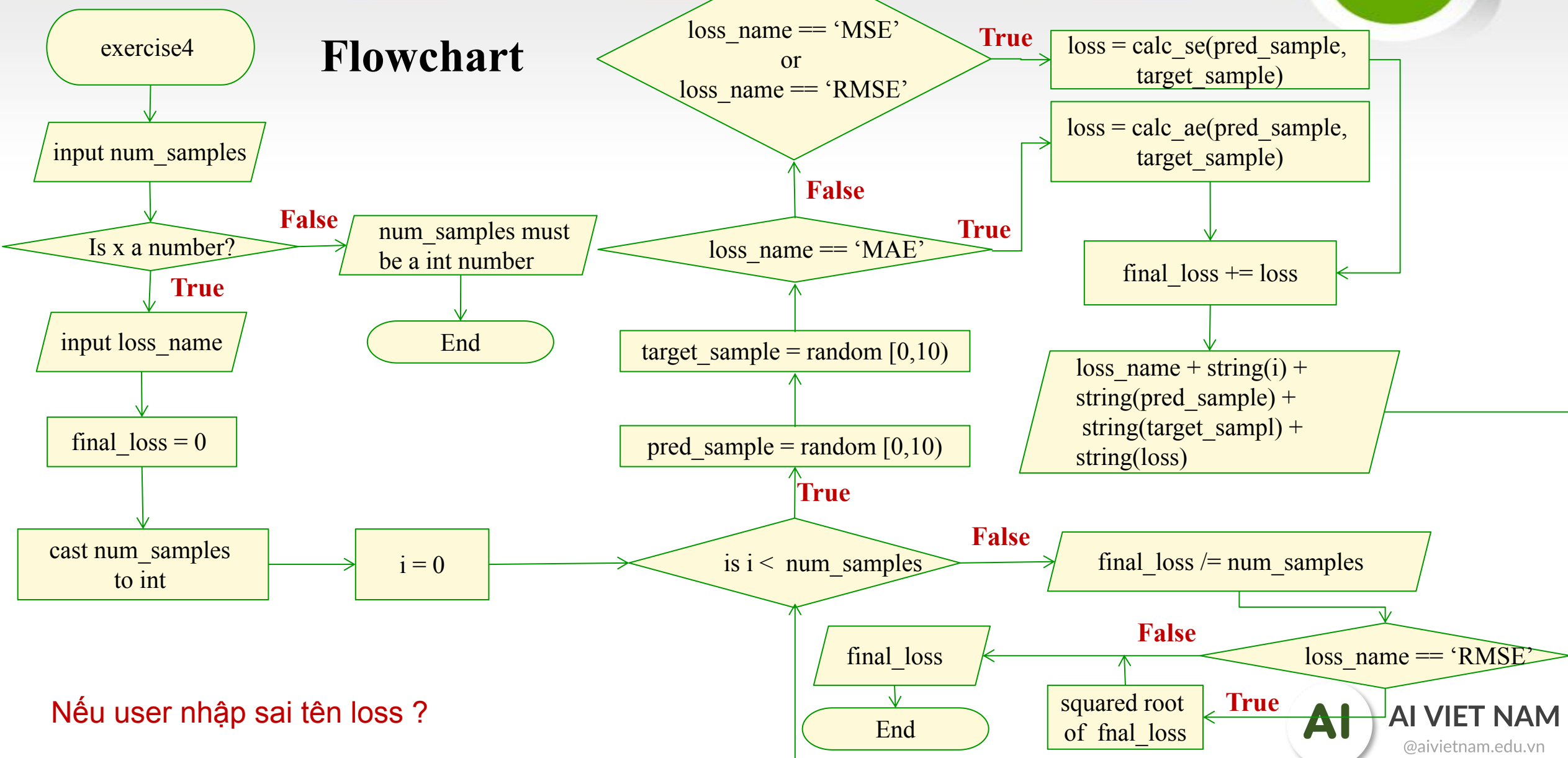
# Exercise4: Viết function tính loss theo regression loss function



# Exercise4: Viết function tính loss theo regression loss function



## Flowchart





- [https://users.csc.calpoly.edu/~jdalbey/SWE/pdl\\_std.html](https://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html)
- <https://www.techgeekbuzz.com/how-to-write-pseudocode/>
- <https://press.rebus.community/programmingfundamentals/chapter/flowcharts/>
- <https://press.rebus.community/programmingfundamentals/chapter/function-examples/>
- <https://press.rebus.community/programmingfundamentals/front-matter/preface/>
- [https://bournetocode.com/projects/GCSE\\_Computing\\_Fundamentals/pages/3-2-7-subroutines.html](https://bournetocode.com/projects/GCSE_Computing_Fundamentals/pages/3-2-7-subroutines.html)



- <https://towardsdatascience.com/pseudocode-101-an-introduction-to-writing-good-pseudocode-1331cb855be7>
- <https://medium.com/@ngunyimacharia/how-to-write-pseudocode-a-beginners-guide-29956242698>





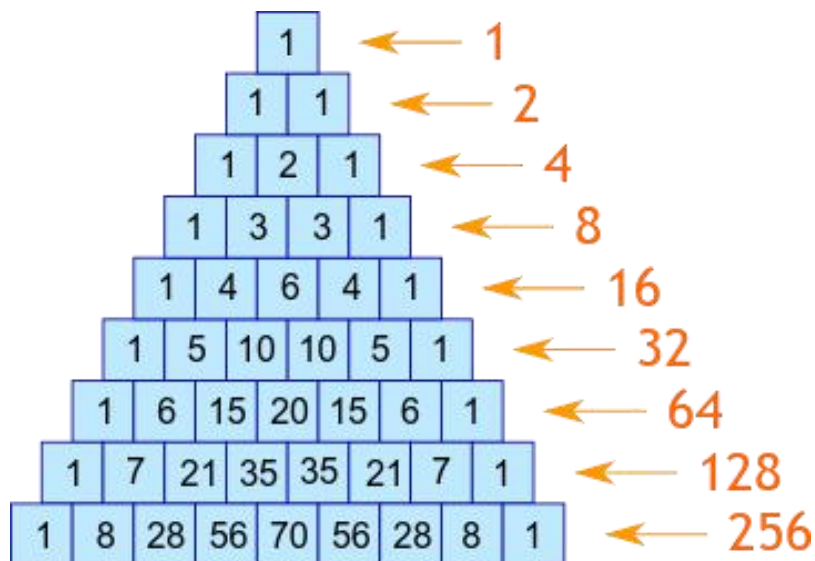
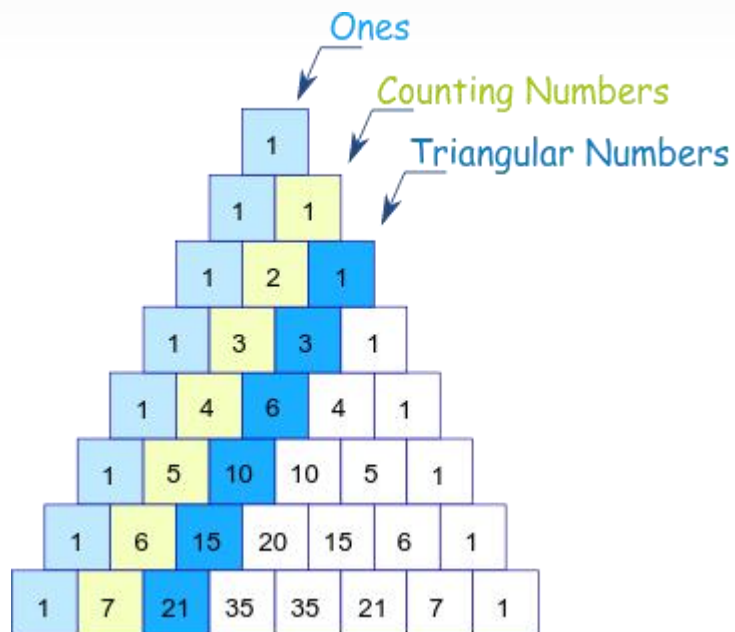


- **Exercise4: Viết function tính loss theo regression loss function**
  - Mean Absolute Error (MAE)
  - Mean Square Error (MSE)
  - Root Mean Square Error (RMSE)
- **Optional Exercises: Giới thiệu idea các bài tập thêm**
  - Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence
  - Exerices6: Viết function xấp xỉ các hàm lượng giác
  - Exercise7: Viết function đảo ngược các vị trí 1 số nguyên dương cho trước

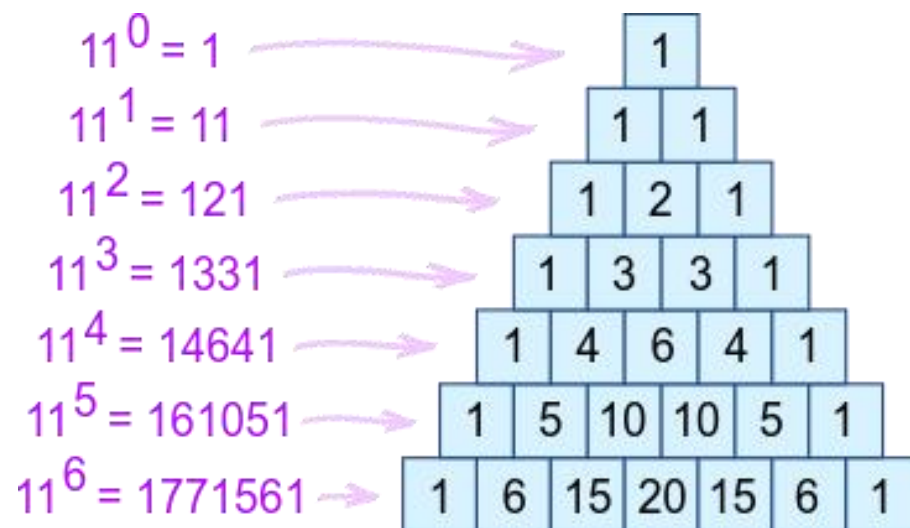
# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence



Symmetrical

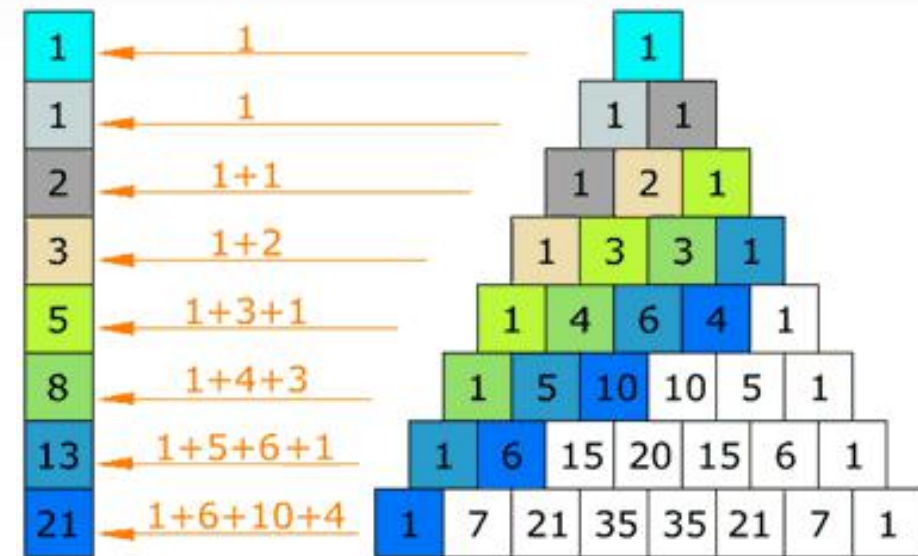
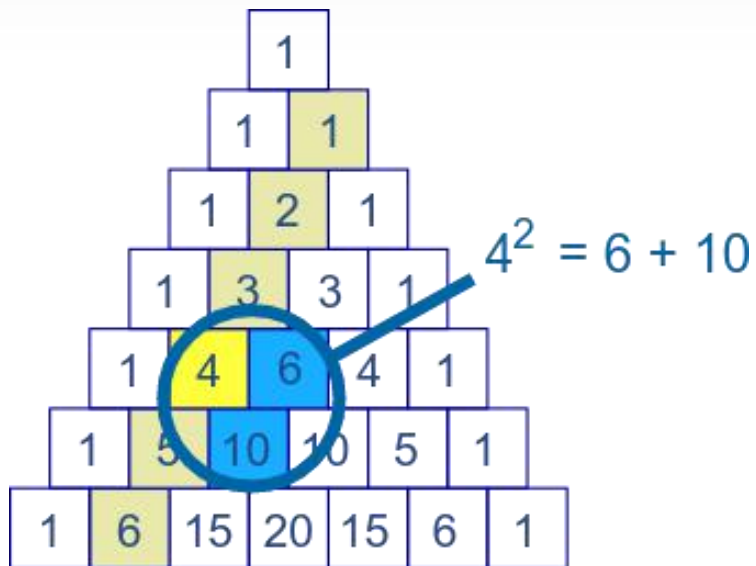


Exponents of 11

# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence



Fibonacci Sequence

# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence

Tosses	Possible Results (Grouped)	Pascal's Triangle
1	H T	1, 1
2	HH HT TH TT	1, 2, 1
3	HHH HHT, HTH, THH HTT, THT, TTH TTT	1, 3, 3, 1
4	HHHH HHHT, HHTH, HTHH, THHH HHTT, HTHT, HTTH, THHT, THTH, TTHH HTTT, THTT, TTHT, TTTH TTTT	1, 4, 6, 4, 1
	... etc ...	

Power	Binomial Expansion	Pascal's Triangle
2	$(x + 1)^2 = 1x^2 + 2x + 1$	1, 2, 1
3	$(x + 1)^3 = 1x^3 + 3x^2 + 3x + 1$	1, 3, 3, 1
4	$(x + 1)^4 = 1x^4 + 4x^3 + 6x^2 + 4x + 1$	1, 4, 6, 4, 1
	... etc ...	

# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence

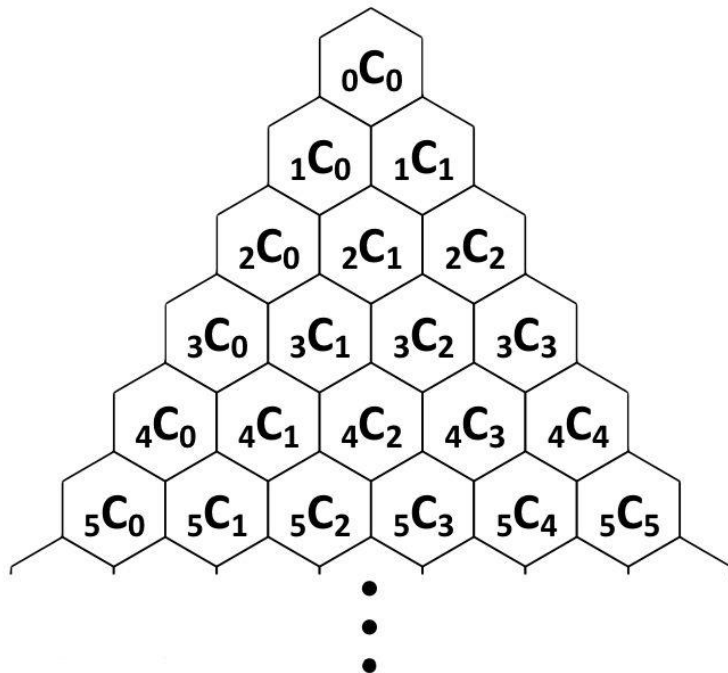
- row index start at 0
- row with  $n$  index:  ${}_nC_0, {}_nC_1, {}_nC_2, \dots, {}_nC_n$
- ${}_nC_r = \frac{n!}{(n-r)!r!}$
- ${}_nC_{r+1} = \frac{n!}{(n-r-1)!(r+1)!}$

$$x! = x(x-1)!$$

$$(x+1)! = (x+1)x!$$

$$(n-r-1)! = \frac{(n-r)!}{n-r}$$

$$(r+1)! = (r+1)r!$$



$$\text{row with } n \text{ index: } {}_nC_{r+1} = \frac{n!(n-r)}{(n-r)!r!(r+1)} = {}_nC_r \frac{n-r}{r+1}$$

# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exercise5: Viết function thực hiện tính Pascal's Triangle và Fibonacci Sequence

Fibonacci sequence = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

n1=0, n2=1:  $0 + 1 = 1$   
n1=1, n2=1:  $1 + 1 = 2$   
n1=1, n2=2:  $1 + 2 = 3$   
n1=2, n2=3:  $2 + 3 = 5$   
n1=3, n2=5:  $3 + 5 = 8$   
n1=5, n2=8:  $5 + 8 = 13$   
n1=8, n2=13:  $8 + 13 = 21$   
n1=13, n2=21:  $13 + 21 = 34$

- set  $n1 = 0, n2 = 1$
- if  $length > 0$ , nếu không dừng chương trình.
- if  $length == 1$ , print 0
- else:
  - loop from 0 to  $length - 1$
  - + print  $n1$
  - +  $temp = n1 + n2$
  - +  $n1 = n2$
  - +  $n2 = temp$





# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exerices6: Viết function xấp xỉ các hàm lượng giác

$$\sin(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{(2n+1)}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$\cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

$$\sinh(x) \approx \sum_{n=0}^{\infty} \frac{x^{(2n+1)}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

$$\cosh(x) \approx \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \frac{x^{10}}{10!} + \dots$$

- Sử dụng loop
- Viết 1 hàm tính giai thừa
- Viết 1 hàm thực hiện phân tử và gọi hàm tính giai thừa



# Optional Exercises: Giới thiệu idea các bài tập thêm



- Exercices6: Viết function xấp xỉ các hàm lượng giác

$$\sin(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{(2n+1)}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$\cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

$$\sinh(x) \approx \sum_{n=0}^{\infty} \frac{x^{(2n+1)}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

$$\cosh(x) \approx \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \frac{x^{10}}{10!} + \dots$$

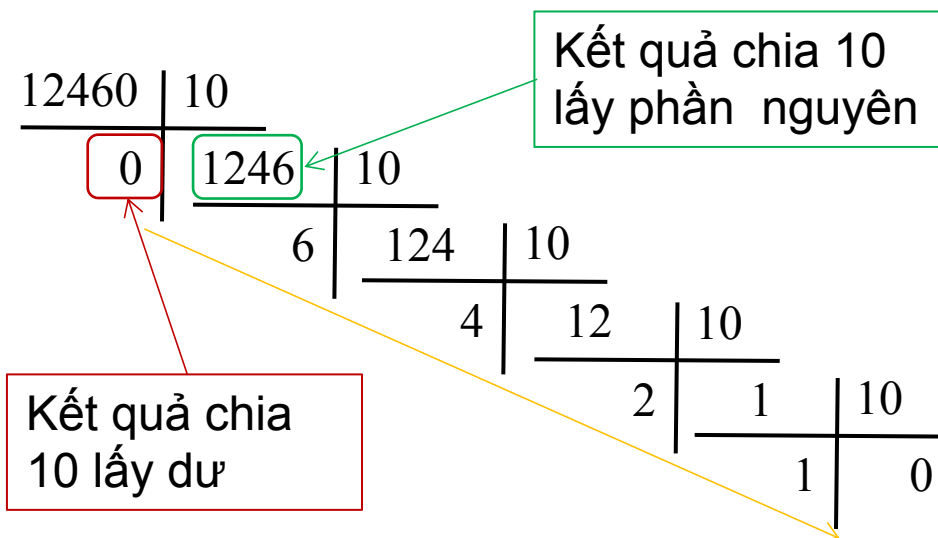
- Sử dụng loop
- Viết 1 hàm tính giai thừa
- Viết 1 hàm thực hiện phân tử và gọi hàm tính giai thừa

# Optional Exercises: Giới thiệu idea các bài tập thêm



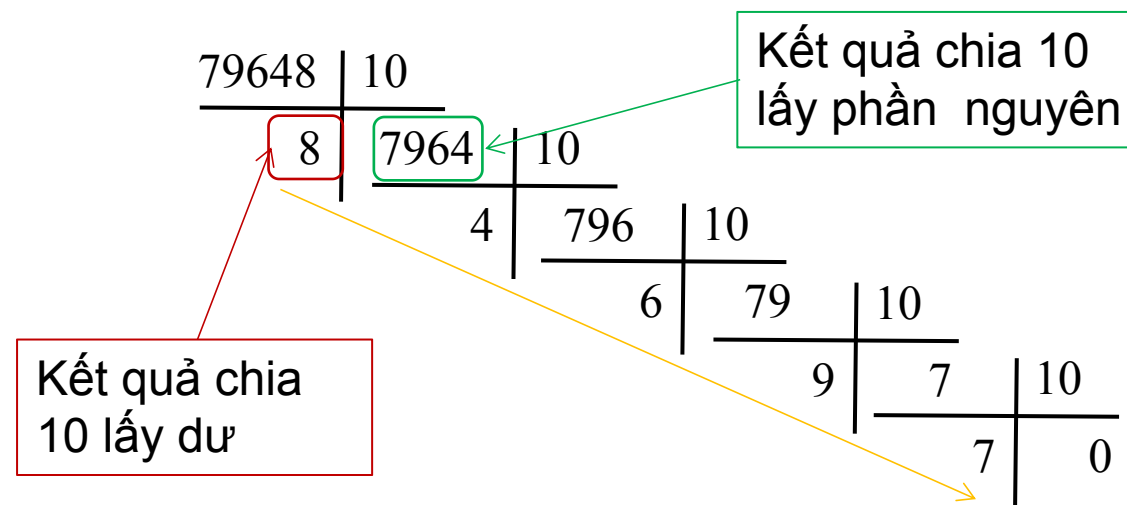
- Exercise7: Viết function đảo ngược các vị trí 1 số nguyên dương cho trước

$$12460 \rightarrow 06421 = 6421$$



$$(((0 \cdot 10 + 6) \cdot 10 + 4) \cdot 10 + 2) \cdot 10 + 1 = 6421$$

$$79648 \rightarrow 84697$$



$$(((8 \cdot 10 + 4) \cdot 10 + 6) \cdot 10 + 9) \cdot 10 + 7 = 84697$$