

## EAST TechSprint Log:

Use this document to log your work for the TechSprint period. Remember, your log should reflect adequate effort to achieve growth in the skills you are learning. This work is to be done **in addition** to any work that you are doing or skills you are developing for your EAST Project (although the work can be complementary). Each log entry should follow the following format:

- Date
- Description of learning goal or task
- Link to tutorial resource(s)
- Example of **your work** based on the tutorial and goals (screenshot preferred), embedded in this document
- Reflection on what you learned, challenges or other important elements

This log will be due by 11:59 PM on the Sunday following the TechSprint period.

---

**Date:** 2/23

**Task:** Learning about 'input' in Python

- Making an algorithm that prints the sum of two integers when getting an input of two integers divided by blank.

**Link to Resource:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%A%EC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23199/%EB%AC%B8%EC%A0%9C0b-%EC%9E%85%EB%A0%A5%EB%B0%9B%EC%95%84%EB%B3%B4%EA%B8%B0>

```

4
5 x = input()
6 my_list = x.split(".")
7 a = int(my_list[0])
8 b = int(my_list[1])
9
10
11 answer = a + b
12
13 print(answer)
14

```

**Reflection:** When using "split()", it is possible to divide a string and put the values into a list.

**Date:** 2/24

**Task:** Learning about the loop in Python

- Making an algorithm to print "Goorm" with a comma between them for certain times when receiving an integer as an input.

**Link to Resource:**

<https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23200/%EB%AC%B8%EC%A0%9C0c-%EB%B0%98%EB%B3%B5%ED%95%B4%EB%B3%B4%EA%B8%B0>

### 1. My Trial

```

4 num = int(input())
5 result = 'Goorm, ' * (num-1) + "Goorm"
6 print(result)
7
8 n = 0
9

```

### 2. An Example Answer

```

n = int(input())
print(", ".join(["Goorm"] * n))

```

**Reflection:** When using "join()", it is easier to insert commas or other elements between the values to make the output easier to be recognized.

**Date:** 2/25

**Task:** Learning about saving data in a list in Python

- When the number of data N(an integer) and N integers are given as inputs, print the N numbers in the reversed order.

**Link to Resource:**

<https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23201/%EB%AC%B8%EC%A0%9C0d-%EC%A0%80%EC%9E%A5%ED%95%B4%EB%B3%B4%EA%B8%B0>

```
n = int(input())

arr = []

for w in input().split():
    t = int(w)
    arr.append(t)

s = [str(w) for w in arr]
s.reverse()
result = " ".join(s)
print(result)
```

**Reflection:**

When getting an input, it is possible to split them using a loop.

It is effective to use reverse() to change the order of the list.

It is possible to use `join()` to a list.

**Date:** 2/27

**Task:** Learning how to make an algorithm to run test cases given as inputs.

- Making an algorithm to print the sum of three given integers that tests certain given cases automatically.

**Link to resource:**

<https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23474/%EB%AC%B8%EC%A0%9C0e-%ED%85%8C%EC%8A%A4%ED%8A%B8%EC%BC%80%EC%9D%B4%EC%8A%A4>

```
def testcase():  
    a, b, c = map(int, input().split())  
    answer = a + b + c  
    print(answer)  
  
tc = int(input())  
for i in range(tc):  
    print("Case #" + str(i + 1))  
    testcase()
```

**Reflection:**

When using mapping, it is possible to set the input data in different variables.

When using functions, it is easier to organize and reuse the same code.

**Date:** 2/28

**Task:** Learning how to create a function in Python

- Making an algorithm that gets two integers by input and prints the bigger value.

**Link to resource:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%A%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/22752/%EB%AC%B8%EC%A0%9C1a-%EC%B5%9C%EB%8C%80%EA%B0%92-%ED%95%A8%EC%88%98>
- <https://github.com/dongyi-kim/IntroductionToAlgorithmProblemSolvingTechniques/blob/master/Chapter01/Problem01A/solution.py>

```
#두 수중에 더 큰 값을 반환하는 함수
def get_max(a, b):
    #[begin]
    if a >= b:
        return(a)
    else:
        return(b)
    #[end]

#두 값을 입력받는다.
p, q = map(int, input().split())
#더 큰 값을 answer에 저장한다
answer = get_max(p, q)
```

**Reflection:**

When making a function, all the cases must be considered not to make an error.

It is effective to use functions when comparing numbers.

**Date:** 3/1

**Task:** Learning how to find the biggest value in an array.

- Making an algorithm that prints out the biggest integer among the N integers given by input.

#### Link to Resource:

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%A8%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/22753/%EB%AC%B8%EC%A0%9C1b-%EB%B0%B0%EC%97%B4%EC%9D%98-%EC%B5%9C%EB%8C%80%EA%B0%92>
- <https://github.com/dongyi-kim/IntroductionToAlgorithmProblemSolvingTechniques/blob/master/Chapter01/Problem01B/solution.py>

```
#배열(data)의 원소 중 가장 큰 정수를 반환하는 함수를 작성해보자
def get_max(data, n):
    #begin
    max_num = data[0]
    for i in range(n-1):
        if data[i+1] >= max_num:
            max_num = data[i+1]
    #end
    return max_num

#데이터의 수를 입력받는다
n = int(input())
#데이터들을 입력받는다
data = list(map(int, input().split()))
#배열의 최대값을 저장한다
answer = get_max(data, n)
#답을 출력한다
print(answer)
```

#### Reflection:

When approaching a map object with integer index([1]...), there is an error that says “map’ object is not subscriptable.” To approach values with index, the variable must be a list or an array.

When setting parameters, it is easier to send input values to the function to be processed.

**Date:** 3/2

**Task:** Learning how to count numbers that meet the given condition by using a loop.

- When N numbers and two certain numbers are given, find how many times do the certain numbers are given in the N numbers.

**Link to resources:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/22884/%EB%AC%B8%EC%A0%9C1c-%EC%B9%B4%EC%9A%B4%ED%8C%85%ED%95%98%EA%B8%B0>
- <https://github.com/dongyi-kim/IntroductionToAlgorithmProblemSolvingTechniques/blob/master/Chapter01/Problem01C/solution.py>

```
def height_detector(heights, n, m, s):
    same_height = 0
    for i in range(n):
        if heights[i] == m or heights[i] == s:
            same_height += 1
    return same_height

n, m, s = map(int, input().split())
heights = list(map(int, input().split()))
answer = height_detector(heights, n, m, s)
print(answer)
```

**Reflection:**

When using functions with parameters, it is necessary to save the result in a variable to print them. When using a loop to add numbers, it is effective to set the variable before starting the loop.

**Date:** 3/3

**Task:** Learning how to count and find the sum of consecutive numbers simultaneously.

**Link to resource:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/22885/%EB%AC%B8%EC%A0%9C1e-%ED%95%A9-%EA%B5%AC%ED%95%98%EA%B8%B02>
- <https://youtu.be/9ca73y0NKok>
- <https://github.com/dongyi-kim/IntroductionToAlgorithmProblemSolvingTechniques/blob/master/Chapter01/Problem01E/solution.py>
- [https://www.learnpython.org/en/String\\_Formatting](https://www.learnpython.org/en/String_Formatting)

1. My trial

```
def weight_calculator(n, p, q, data):
    possible_weight_total = 0
    possible_num = 0
    for i in range(n):
        if data[i] <= p:
            possible_weight_total += data[i]
            possible_num += 1

    if possible_weight_total <= q:
        y_or_n = "YES"
    else: y_or_n = "NO"

    result = str(possible_num) + " " + str(possible_weight_total) + "\n" + y_or_n
    return(result)

n, p, q = map(int, input().split())
# n = the number of people in the club
# p = a weight limitation for each person
# q = a total weight limitaion
data = list(map(int, input().split()))

answer = weight_calculator(n, p, q, data)
print(answer)
```



## 2. Revised version

```
def weight_calculator(n, p, q, data):
    possible_weight_total = 0
    possible_num = 0
    for i in range(n):
        if data[i] <= p:
            possible_weight_total += data[i]
            possible_num += 1

    result = "%d %d" % (possible_num, possible_weight_total)
    #result = str(possible_num) + " " + str(possible_weight_total) + "\n" + y_or_n
    print(result)

    if possible_weight_total <= q:
        print("YES")
    else:
        print("NO")

n, p, q = map(int, input().split())
# n = the number of people in the club
# p = a weight limitation for each person
# q = a total weight limitaion
data = list(map(int, input().split()))

weight_calculator(n, p, q, data)
```

### Reflection:

When creating a string that contains different types of variables, such as integers, it is effective to adopt "string formatting."

A function can be used to print certain values as well as to return the values.

**Date:** 3/4

**Task:** Find the sum of consecutive data using the loop

- Making an algorithm to print the sum of integers when N integers are given consecutively, distinguished by each space.

**Link to Resource:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23602/%EB%AC%B8%EC%A0%9C1d-%ED%95%A9-%EA%B5%AC%ED%95%98%EA%B8%B01>
- <https://github.com/dongyi-kim/IntroductionToAlgorithmProblemSolvingTechniques/blob/master/Chapter01/Problem01D/solution.py>

```
def sum_data(n, data):
    result = 0
    for i in range(n):
        result += data[i]

    return result

n = int(input())
data = list(map(int, input().split()))

answer = sum_data(n, data)
print(answer)
```

**Reflection:**

When processing the consecutive data, it is effective to split them into a list, and then put them into a function.

**Date:** 3/4

**Task:** Learning how to get the location of a certain value when consecutive data is given.

- An array consists of N integers and the value M is given. Make a program that prints the index where M exists in this array. (If M is not in the array, print -1.)

**Link to Resource:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23603/%EB%AC%B8%EC%A0%9C1f-%ED%83%90%EC%83%89%ED%95%98%EA%B8%B01>

```
def m_detector(n, m, data):
    answer = -1
    for i in range(n):
        if data[i] == m:
            answer = i
    return answer

n, m = map(int, input().split())
data = list(map(int, input().split()))

result = m_detector(n, m, data)
print(result)
```

**Reflection:** In Python, when checking whether the two values are the same, the equality operator (==) is used. "=" is an assignment operator used to assign a value.

**Date:** 3/6

**Task:** Learning how to get the location of a certain value when consecutive data is given. -2

- For certain data that meet the given condition, get locations of only the first and last data and print them.

**Link to Resource:**

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/22886/%EB%AC%B8%EC%A0%9C1g-%ED%83%90%EC%83%89%ED%95%98%EA%B8%B02>
- [https://www.w3schools.com/python/ref\\_string\\_join.asp](https://www.w3schools.com/python/ref_string_join.asp)

```

n = int(input())

first_index = -1
last_index = -1
for i in range(1, n+1):
    name = input()

    if name == "AJOU":
        last_index = i
        if first_index == -1:
            first_index = i

print(first_index, last_index)

```

### Reflection:

To make it easier to determine whether a value meets certain conditions, it is useful to set initial variables before using "if."

"join()" only works for strings.

When printing values with the form of (value1, value2), the values are printed in the same line with a space between them. The result of the code is "value1 value2."

**Date:** 3/7

**Task:** Practicing writing codes that suggest challenging conditional statements.

- N numbers of integers are given as numeral data. Find the data that is closest to the mean of overall data, and print its value and its "own number." "Own number" of each data is given from 1 to N by its order. If there is more than one data that meets the condition, the data in which its "own number" is low takes priority.

### Link to resource:

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%A C%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/les son/22887/%EB%AC%B8%EC%A0%9C1h-%ED%83%90%EC%83%89%ED%95%98%EA%B8%B03>

- [https://www.w3schools.com/python/ref\\_func\\_min.asp](https://www.w3schools.com/python/ref_func_min.asp)

```
n = int(input())
data = list(map(int, input().split()))
mean = sum(data) / n
abs_dev_list = []

for i in range(n):
    abs_dev = abs(data[i] - mean)
    abs_dev_list.append(abs_dev)

com_value = abs_dev_list[0]
own_num = 1
my_num = data[0]
for j in range(n):
    if abs_dev_list[j] < com_value:
        com_value = abs_dev_list[j]
        my_num = data[j]
        own_num = j + 1

print(own_num, my_num)
```

### Reflection:

“sum()” can be used to find the sum of values of a list.

“abs()” is used to find an absolute value of a number.

**Date:** 3/7

**Task:** Learning how to create a Selection Sort

### Link to Resource:

- <https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%A%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/22889/%EB%AC%B8%EC%A0%9C1i-%EC%84%A0%ED%83%9D%EC%A0%95%EB%A0%AC-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0>

```

n = int(input())
data = list(map(int, input().split()))

for i in range(n):
    data_2 = data[i:]
    min_num = min(data_2)
    idx = data.index(min(data_2))
    data[i], data[idx] = data[idx], data[i] #swap

for j in range(n):
    print(data[j], end = ' ')

```

### Reflection:

When exchanging values in a list, it is useful to “swap” two values by exchanging their index.

“min()” is used for finding the minimum value of a list.

**Date:** 3/7

**Task:** Learning how to make a code to calculate the sum of the sum of sequences

- Find the sum of 1, 1+2, 1+2+3, ..., 1+2+3+...+n

### Link to Resource:

<https://edu.goorm.io/learn/lecture/554/%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EB%AC%B8%EC%A0%9C%ED%95%B4%EA%B2%B0%EA%B8%B0%EB%B2%95-%EC%9E%85%EB%AC%B8/lesson/23047/%EB%AC%B8%EC%A0%9C1j-%ED%95%A9-%EA%B5%AC%ED%95%98%EA%B8%B03>

```
#Find the sum of 1, 1+2, 1+2+3, ..., 1+2+3+...+n
```

```
n = int(input())

final = 0
for i in range(1, n+1): #i이하
    first = 0
    for j in range(1, i+1): #1이상 i 이하
        first += j
    final += first
print(final)
```

### Reflection:

When setting range on "for loop," "range(a, b+1)" means "from a to b."

It is easy to find the sum of sequences when using "for loop" in Python.