Chapter 7.

List, Tuples and matplotlib

Starting out with Python

Kyu Lee. Ph. D.

Computer Science

List

Basic usage of list

- How to use list
  - num = [10, 20, 30]
- How to access to the single individual element
  - print (num[1]) # using the index
- What happen if we use out of index
  - num[3] # error
  - num = []
  - num[0] = 20

Basic usage of list

- Create a list with the values

```
numbers = [5, 10, 15, 20]
print (numbers)
```

- Create a list with range function

```
numbers = list(range(5))
print (numbers)
```

- Create a list with Repetition Operator

```
numbers = [0] * 5

print (numbers)
```

```
numbers = [0, 1, 2] * 3

print (numbers)
```

List with a for loop

- List as an iterator

```
numbers = [10, 20, 30, 40, 50]
for n in numbers:
       print (n, end=' ')
```

- Indexing

```
numbers = [10,20,30,40,50]

for i in range(5):
       print (numbers[i] , end='\t')
print ()

for i in range(0, 5, 2):
       print (numbers[i] , end='\t')
```

Change list values

- Let’s think of two cases

```
num = [10, 20, 30, 40, 50]

for i in range(5):
        num[i] += 10

for v in num:
        print (v, end=' ')
```

```
num = [10, 20, 30, 40, 50]

for v in num:
        v += 10
        print (v, end=' ')      # Values are added to 10
for v in num:
        print (v, end=' ')
```

Values are not changed. Why?

Create a list with the 5 user inputs

- 5 User inputs

```
N = 5
numbers = []
for i in range(5):
       user input = int(input('Enter your input'))
       numbers.insert(i, user_input)

for v in numbers:
       print (v, end=' ')
```

- 5 random values

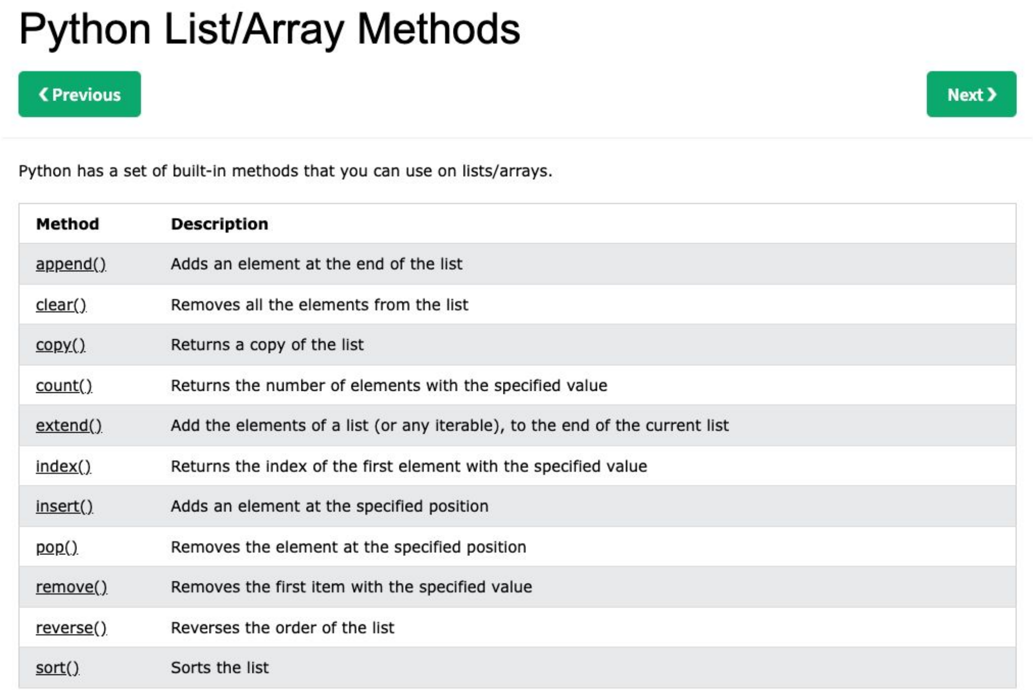
```
import random
N = 5
randnums = []
for i in range(N):
       randnums.insert(i, random.randint(0,100))

for v in randnums:
       print (v, end=' ')
```

Exercise 1

- Write a Python function `getSum(numbers)` that calculates the sum of a given list of numbers, excluding the minimum and maximum values from the list. Your function should **return** the resulting summation.
- For example, given the list `[3, 8, 2, 6, 1, 9, 4, 7]`,
    - the minimum value is 1 and the maximum value is 9.
    - The function should calculate the sum of the remaining numbers `[3, 8, 2, 6, 4, 7]`, which totals 30 (3 + 8 + 2 + 6 + 4 + 7).
- Example
    - numbers = [10,25, 15,35, 50]
    - Output
        - 75
    - In this example, the min value is 10 and max is 50. Thus the total except min and max is 75

Python List/Array methods

- Document Link
  - https://docs.python.org/3/tutorial/datastructures.html#more-on-lists



Python List/Array Methods

Python has a set of built-in methods that you can use on lists/arrays.

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

min( ) and max( ) built-in function

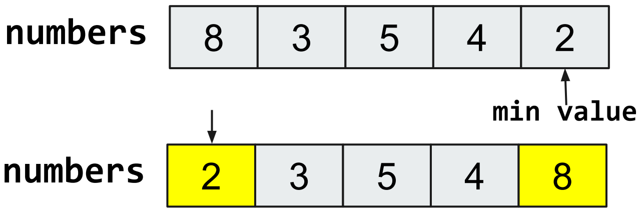
- min() and max()
  - built-in function

```
nums = [15, 5, 10, 25, 20]


print (min(nums))
print (max(nums))
```

Exercise 2

Write a Python program that includes two functions: `genRandom(N)` and `findMin(numbers)`.

1. The `genRandom(N)` function should generate a list of random numbers between 0 and 100, inclusive, and **return** the list as the output. The length of the list should be determined by the parameter N
2. The `findMin(numbers)` function should take a list of numbers as input and find the minimum value in the list. It should then move this minimum value to the index 0 of the list.
    a. Once the minimum value is identified, it should be **swapped** with the **first** element of the list.
    b. No return value.



numbers | 8 | 3 | 5 | 4 | 2

min value

numbers | 2 | 3 | 5 | 4 | 8

- Use the **built-in** function
    - min( ) to find the smallest value in the list
- Use the **method**
    - index( ) to get the index of the smallest element

Finding Items in Lists

- in Operators

```
num_list = [10, 20, 30, 40, 50]

target = int(input('Enter the target number'))
if target in num list:
        print ("Target found in the list")
else:
        print ("Target is not in the list")
```

```
num_list = [10, 20, 30, 40, 50]

target = int(input('Enter the target number'))

if target in num_list:
        print ("Target found at the index ", num_list.index(target))
else:
        print ("Target is not in the list")
```

Finding Items in Lists

- Traditional way to find an item

```
num_list = [10, 20, 30, 40, 50]

target = int(input('Enter the target number'))

for i in range(len(num_list)):
    if num_list[i] == target:
        print ("Target found at the index ", i)
        break
if ( i == len(num_list)-1):
    print ("Target is not in the list")
```

Append

- Append an item

```
num1 = [10, 20, 30, 40, 50]
num1.append(60)
num1
```

```
num1 = []
num1[0] = 10   # ERROR
```

append with index ?

- Append a list

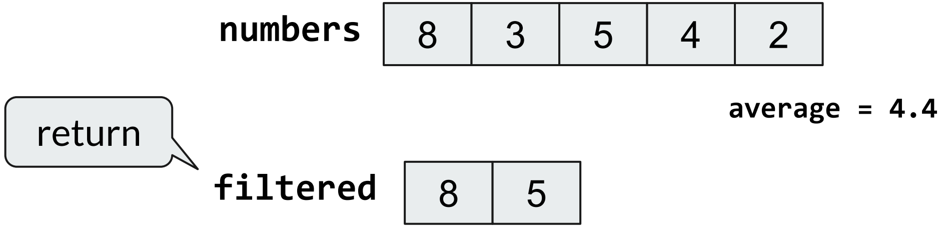
```
num1 = [10, 20, 30]
num2 = [40, 50, 60]

num1.append(num2)    # Not [10, 20, 30, 40, 50, 60]
num1
```

number of elements num1 : 4 why?

Exercise 3

Write a Python program that includes two functions: `genRandom(N)` and `filterAvg(numbers)`.

1. The `genRandom(N)` function should generate a list of random numbers between 0 and 100, inclusive, and **return** the list as the output. The length of the list should be determined by the parameter **N**
2. The `filterAvg(numbers)` function should take a list of numbers as parameter and **find the elements which are greater than average**. It should return the result values as a **list**.
    - ○ use the built-in functions
        - ○ sum(), len( )



numbers | 8 | 3 | 5 | 4 | 2

average = 4.4

return

filtered | 8 | 5

- Example
    - ○ numbers = [1,2,3,4,5]
    - ○ Output: 4 5
- You can use any methods in list
    - ○ https://docs.python.org/3/tutorial/datastructures.html#more-on-lists

Take the multiple numbers in a string

- Input all numbers at once as a string value
  - input_str = input( )   # 10 20 30 40 50
- Split the string list into the string list
  - numbers = input_str.split()
- Convert each element to the integer value
  - for i in range(len(numbers)):
    - numbers[i] = int(numbers[i])

Take multiple user input in a line

Input multiple values in a line

Basic Code

Split input values by the space ‘ ‘

```
uservalues = input( )         # 1 2 3 4 5
numbers = list(uservalues.split())
print (numbers)               # [‘1’, ‘2’, ‘3’, ‘4’, ‘5’]

for i in range(len(numbers)):
     numbers[i] = int(numbers[i])
print (numbers)
```

Change the letter ‘1’ to integer 1

Shortened Code

```
numbers = list(map(int, uservalues.split()))
print (numbers)
return numbers
```

The way to input all values in a line and make it as a list

Exercise 4

- Write a function `getInput( )` that prompts the user to enter multiple values in a single line.
  - The function should then split these values and convert them into a list of **integers**, and return it.
  - See the slide page 17

- Given two lists of numbers, write a function `listSum(list1, list2)` to create a new list that contains the summation of two values within each list for elements at the same index.
  - We assume that the length of two lists are same
  - For example, if the two lists are `[1, 2, 3]` and `[4, 5, 6]`, the new list should be `[5, 7, 9]`.
  - The function should take two lists as parameters and **return** a new list as output.

list1	1	3	5	4	2	7	8	1	2	5
	+									
list2	7	8	1	2	5	1	3	5	4	2
	=									
result list	8	11	6	6	7	8	11	6	6	7

return this

Extend()

- Append a list

```
num1 = [10, 20, 30]
num2 = [40, 50, 60]


num1.extend(num2)    # [10, 20, 30, 40, 50, 60]
num1
```

count()

- The number of elements with the specified value

```
num1 = [0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1]

num1.count(1)  # number of occurrences
```

Exercise 5

```
# Test program will give you multiple integer values in a line
# Your program should take entire values at once
```

- Write a function `getInput( )` that prompts the user to enter multiple values in a single line.
  - The function should then split these values and convert them into a list of **integers**, and return it.
  - See the slide page 17
- Write a function `findMost(numbers)` that finds the number that occurs most frequently in the list numbers.
  - return the most frequently occurred number
  - use the **count()**
- **Return value: the element that occurs most frequently**

return this

numbers	1	2	2	4	2

the occurrences of 2 = 3

The most frequently used number = 2
return 2

pop()

- Removes the element at the specified position

```
num1 = [10, 20, 30]
num1.pop(len(num1)-1)


# Try this
num1.pop()
```

insert()

- insert an item into the list
  - need an index and value

```
num_list = [10, 20, 30, 40, 50]

add_val = 15

num_list.insert(0, add_val)

for v in num_list:
       print (v, end=' ')
```

insert()

- insert an item with out-of-range index

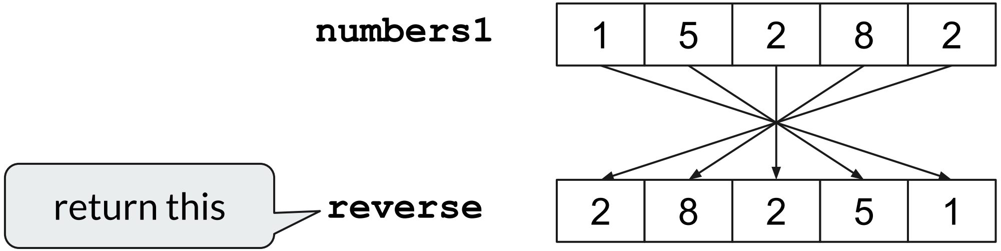
```
num_list = [10, 20, 30, 40, 50]

num_list.insert(10, 60)   # do not try with uncertain index
print (num_list)
```

```
num_list = [10, 20, 30, 40, 50]

num_list[5] # Index Error
```

Exercise 6

- Write a function `getInput( )` that prompts the user to enter multiple values in a single line.
    - The function should then split these values and convert them into a list of integers, and return it.
    - See the slide page 17

- Write a function `makeReverse(numbers)` that takes a list of numbers as input and **returns** a new list with the numbers in reverse order.
    - For example, if the input list is `[1, 2, 3, 4, 5]`, the output list should be `[5, 4, 3, 2, 1]`.
- Do <u>not</u> use reverse(). Implement your own algorithm
- Use the methods
    - pop(), insert( ) or append( )
        - pop() from original list and insert() or append() to the new list.
        - It reverses the order of the list

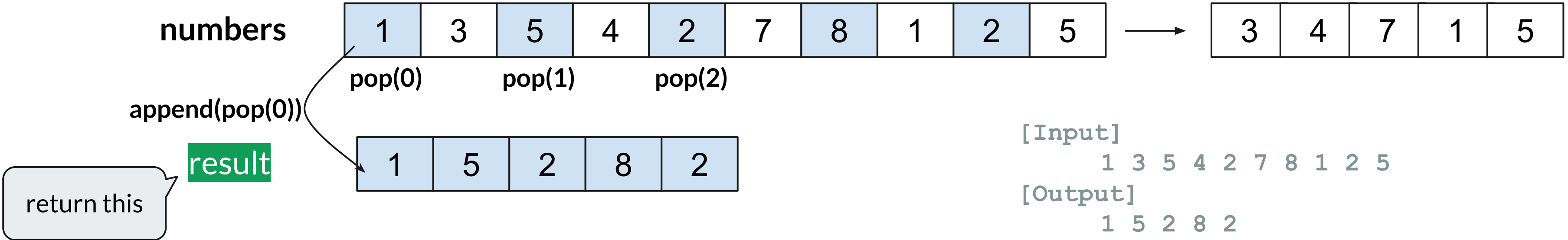


```
[Input]
     1 5 2 8 2
[Output]
     2 8 2 5 1
```

Exercise 7: Even Index Elements [0], [2], [4], [6], …

- Write a function `getInput( )` that prompts the user to enter multiple values in a single line.
  - The function should then split these values and convert them into a list of integers, and **return** it(list of integers).
  - See the slide page 17

- Write a function `evenList(numbers)` that takes a list of numbers as input and **returns** a new list with the numbers at the **even index**, including index 0.
  - For example, if the input list is `[11, 13, 15, 17, 19]`, the output list should be **`[11, 15, 19]`**
  - At this time, the **original list** should be [13, 17] excluding the numbers at the even index from original list
  - **return** the output list [1,3,5]
- Use the methods
  - pop(), insert( ) or append( )
  - After copying the even elements from the original list “numbers”, it should have only odd elements.



numbers 1 3 5 4 2 7 8 1 2 5 → 3 4 7 1 5

result 1 5 2 8 2

```
[Input]
    1 3 5 4 2 7 8 1 2 5
[Output]
    1 5 2 8 2
```

Insert an item into sorted list

- Find the index to be inserted

```
num_list = [10, 20, 30, 40, 50]

add_val = 25

flag = 0
for i in range(len(num_list)):
       if ( num_list[i] > add_val):
               num_list.insert(i, add_val)
               flag = 1
               break
if ( flag == 0 ):
       num_list.insert(i+1, add_val)

for v in num_list:
       print (v, end=' ')
```

Insert an item into sorted list : version 2 (for - else)

- for else

```
num_list = [10, 20, 30, 40, 50]

add_val = 75

for i in range(len(num_list)):
    if ( num_list[i] > add_val):
        num_list.insert(i, add_val)
        break
else:
    num_list.insert(i+1, add_val)

for v in num_list:
    print (v, end=' ')
```

Insert an item into sorted list : version 3 (while - else)

- while - else

```python
num_list = [10, 20, 30, 40, 50]

add_val = 65

i = 0
while i < (len(num_list)):
       if ( num_list[i] > add_val):
               num_list.insert(i, add_val)
               break
       i += 1
else:
       num_list.insert(i+1, add_val)

for v in num_list:
       print (v, end=' ')
```

Sort() method of list

- sort( )

```
import random


N = 5
rdnums = []
for i in range(N):
       rdnums.insert(i, random.randint(0,100))


print (rdnums.sort())
```

Sorted() built-in function

- sorted( )

```
nums = [15, 10, 5, 25, 20]

sorted_nums = sorted(nums)

print (nums) # 15 10 5 25 20
print (sorted_nums) # 5 10 15 20 25
```

Exercise 8

- Write a function `getInput( )` that prompts the user to enter multiple values in a single line.
  a. The function should then split these values and convert them into a list of integers, and **return** it.
  b. See the slide page 17

- Write a function `insertOne(numbers)` that takes the list `numbers` as a **parameter**, and one integer value `val` as user **input** and **then** insert the value `val` to the list `numbers` `at the correct index to keep the list sorted,` assuming that the list `numbers` is **already sorted.**
  a. For example, if the list `numbers` is `[1, 2, 4, 5]`, and the insert value `val` is 3,
     - the output should be [1,2,3,4,5].
     - No return value
  a) Use the insert() method. Find the index to insert the element.

no return value. Use the same list

- Requirement
  a) do not use any sort() function // will fail to pass the test in Github Classroom

numbers	10	15	25	30	35	
insertion value	20					
numbers	10	15	20	25	30	35

```
Input
10 15 25 30 35
20
Output
10 15 20 25 30 35
```

remove()

- remove with **value**

```
nums = [10, 20, 25, 30, 35]
delete_val = int(input('Enter a number'))
nums.remove(delete_val) # nums.remove(40) => ValueError
nums
```

```
nums = [10, 20, 25, 30, 35]

# try remove with invalid value

nums.remove(50)
```

Value error !

remove() with try-except

- remove with **value**

```
nums = [10, 20, 25, 30, 35]
try:
       delete_val = 40                 # 40 is not in the list
       nums.remove(delete_val)         # Cause an error since the delete key
value is not in the list
except ValueError:
       print ('Value Error: There is no value ', delete_val)


    Key Error: There is no value  40
```

del statement

- **del** statement
  - The del keyword is used to delete objects. In Python everything is an object, so the del keyword can also be used to delete variables, lists, or parts of a list etc.

```
nums = [10, 20, 25, 30, 35]


del nums[0]                     # remove the 1st element
nums



i = 10
del  i           delete an object i
```

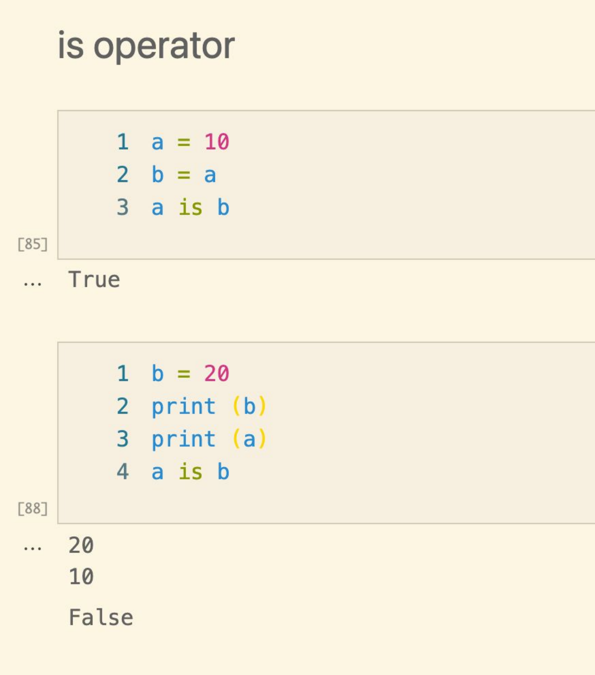
reverse( )

- reverse( )
    - without any argument
    - make a reverse order list

```
nums = [1, 2, 3]

nums.reverse()

nums
```

is operator

- Are two objects are same or not?



is operator

```
[85]
a = 10
b = a
a is b
```

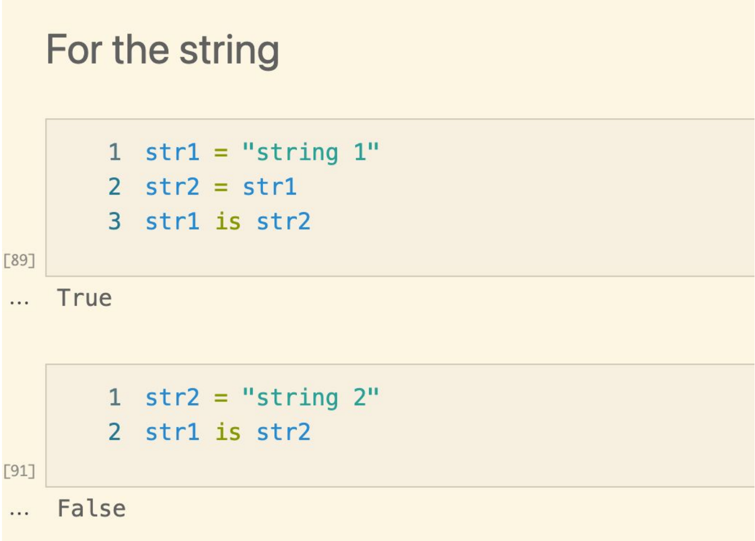
```
True
```

```
[88]
b = 20
print (b)
print (a)
a is b
```

```
20
10
False
```

is operator

- For the string



For the string

```
[89]
str1 = "string 1"
str2 = str1
str1 is str2
```

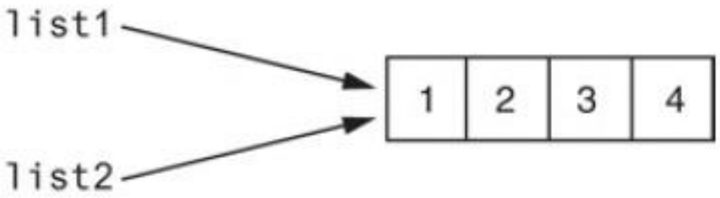
```
... True
```

```
[91]
str2 = "string 2"
str1 is str2
```

```
... False
```

Copying lists

- When we assign a list to another list
  - for example,



```
l1 = [10, 20, 30]
l2 = l1
```

```
l2 is l1
true
```

```
l2[0] = 100
print (l1) # 100 20 30
```

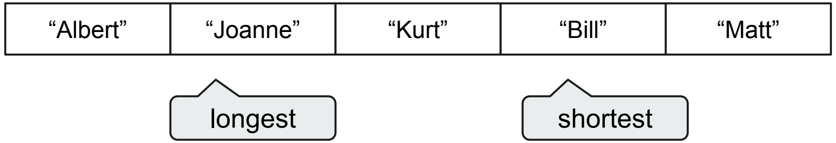
```
l2 = [40, 50, 60]
l2 is l1 # false
```

Exercise 9

- make a function `findNames(names)` that takes a list of names as a parameter and find the longest and shortest names in the given list names
  - For example, in the list of name [Albert, Joanne, Kurt, Bill, Matt]
  - Find the shortest name and longest name based on the **string length and then alphabetical order**
    - Shortest name: Bill
      - if there are multiple names that has the same length, find the least name in **alphabetical order**.
      - For example, 'Bill' is the shortest name since it is the least in alphabetical order ['Kurt', 'Bill', 'Matt'].
    - Longest name: Joanne
      - if there are multiple names that has the same length, find the greatest name in **alphabetical order**.
  - Return multiple items
    - **return longest, shortest**

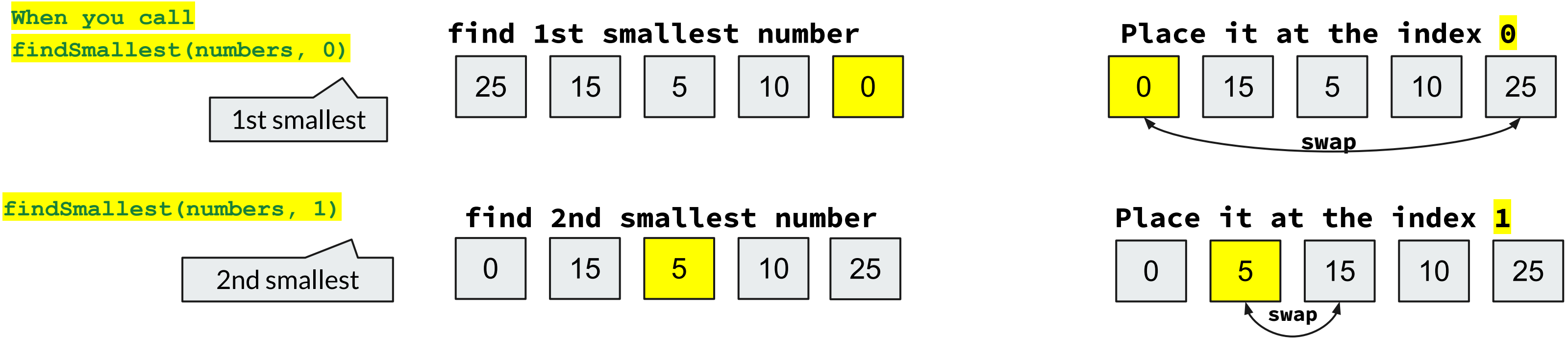
return these

```
Input
Albert Joanne Kurt Bill Matt

Output
Bill Joanne
```



Exercise 10 Find the i_th smallest value

- Write a function `findSmallest(numbers, i)` that takes a list of **numbers** and an index `i` as parameters. The function should find the (`i+1)_th` smallest number in the list and place it at index i position in the list. The value i can be from 0 to N-1,where N is the number of elements
- For example,



When you call
findSmallest(numbers, 0)

1st smallest

find 1st smallest number

25	15	5	10	0

Place it at the index 0

0	15	5	10	25

swap

findSmallest(numbers, 1)

2nd smallest

find 2nd smallest number

0	15	5	10	25

Place it at the index 1

0	5	15	10	25

swap

- Return value: **None**
- Do **not** use sort(), sorted(), or any sort functions.
- Assumption:
  - when you call findSmallest(numbers, **i**),
    - **all elements from index 0 to i-1 are already sorted, and**
    - **they are less than (i+1)_th smallest number**

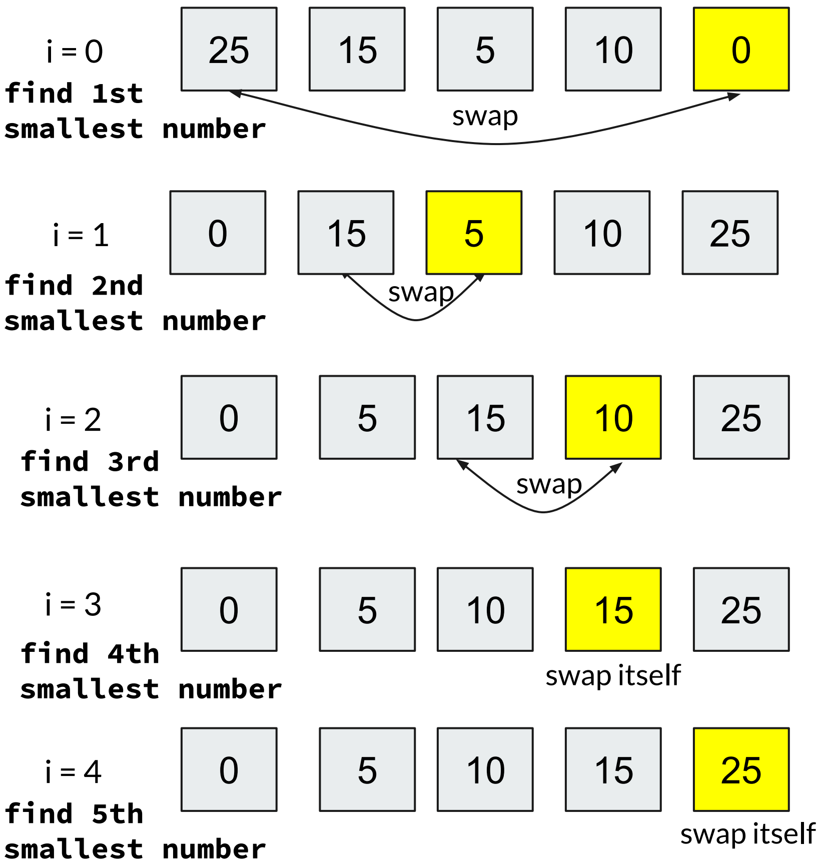
Exercise 10

- Make a function `selectionSort(number)` that takes a list of numbers and call `findSmallest(numbers, i)` N times with the value i from 0 to N-1, where `N` is the number of elements in the list.

- This function should call the `findSmallest` function `N` times,
  - `findSmallest(numbers, i) # i will be from 0 to N-1`

- `Return value:` **None**

Example



i = 0 **find 1st smallest number**: 25 15 5 10 0 — swap

i = 1 **find 2nd smallest number**: 0 15 5 10 25 — swap

i = 2 **find 3rd smallest number**: 0 5 15 10 25 — swap

i = 3 **find 4th smallest number**: 0 5 10 15 25 — swap itself

i = 4 **find 5th smallest number**: 0 5 10 15 25 — swap itself

Assignments

Introduction to Python Programming

Practice: in line for loop

- Make a list from the another list

```python
l1 = [10, 15, 20, 25, 30, 35]

l2 = [v for v in l1 if v % 2 == 0]
print (l2) # [10, 20, 30]
```

```python
l1 = [10, 15, 20, 25, 30, 35]

l2 = [i for i in range(len(l1)) if l1[i] > 20 ]
print (l2)
```

```python
are = ['a', 'r', 'e']
idxlst = [ 'assertive'.find(are[j]) for j in range(len(are))]
print (idxlst)
```

Instruction to submit your work

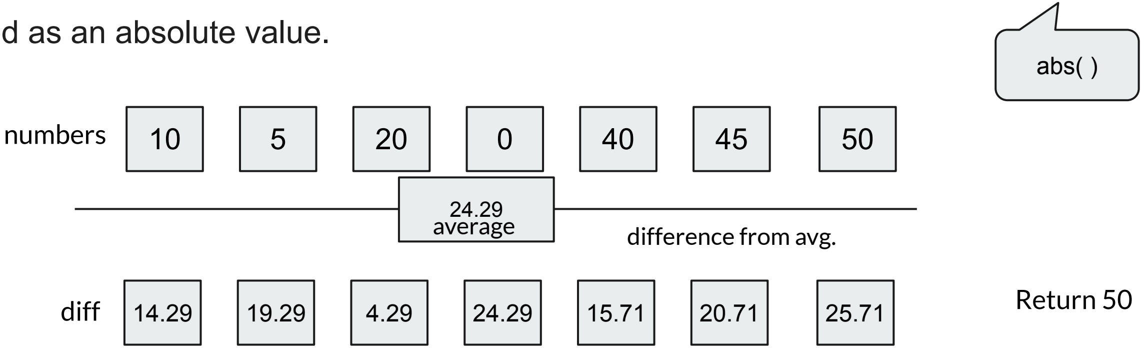
Make multiple commits, even if there are small changes.

It is strongly recommended to increase the **frequency of commit/push actions**

- When you accept the Classroom Assignment through the link, it is the beginning time of the question.
- After initializing the Repository, you should **at least commit/push every 5 minutes.**(frequent commits/push)
    - A major part has been built (e.g., for loop / if statement)
    - When you meet errors, try to fix errors,
- This will allow me to see your progress and how you have worked on your code
- This commit log can make us see the program's progress and prevent **plagiarism**.
    - **More commits/push, more points**
    - **Only one commit, no point.**

Assignment 7-1

- Write a function called getFarNumber(numbers) that takes a list of numbers as a parameter from main function and returns the number which has the greatest gap from the average. The gap will be considered as an absolute value.



abs( )

numbers	10	5	20	0	40	45	50
diff	14.29	19.29	4.29	24.29	15.71	20.71	25.71

24.29
average

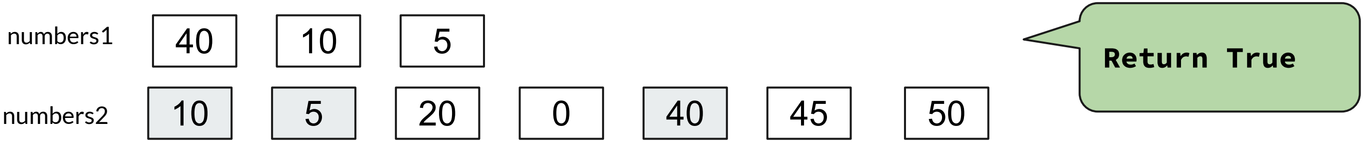
difference from avg.

Return 50

- Example: `10 5 20 0 40 45 50 55 9 10`
- Return value
  - `55`

Assignment 7-2 : Determine the sub-list

- Write a function `isSubList(numbers1, numbers2)` that takes two lists of numbers and decide the list *`numbers1`* is the sublist of *`numbers2`*.
    - `Sublist: Each elements in` *`numbers1`* `is also an element of` *`numbers2`*
- Requirement
    - **Do not use "in" operator** to practice "for loop" or "while loop" with the list values
- Return value: True or False



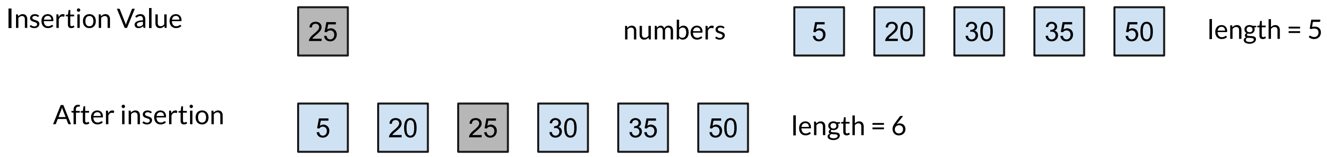
numbers1	40	10	5				
numbers2	10	5	20	0	40	45	50

`Return True`

Assignment 7-3 : Insert an element into the list

- Write a function `insertOne(numbers, val)` that takes the list `numbers`, and one integer value `val` as **parameters** and **then** insert the value `val` to the list `numbers` `at the correct index to keep the list sorted,` assuming that the list `numbers` is **already sorted.**
    - For example, if the list `numbers` is `[1, 2, 4, 5]`, and the insert value `val` is 3,
        - Find the insertion position, and use the method of the list `insert( )`
        - the numbers should be [1,2,3,4,5].
        - No return value

no return value



Insertion Value 25

numbers 5 20 30 35 50 length = 5

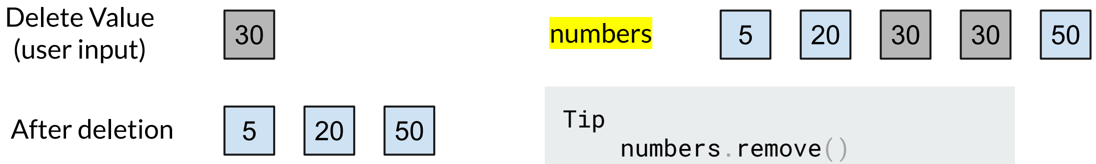
After insertion 5 20 25 30 35 50 length = 6

- Do not use sort() or sorted() functions

`Do not use any sort() functions`

Assignment 7-4 : delete an element from the list

- Write a function **`deleteOne(numbers, value)`** that takes a list of number and a value for deletion from the list. The function should then delete all occurrences of number **`value`**.



Delete Value (user input) 30

numbers 5 20 30 30 50

After deletion 5 20 50

```
Tip
   numbers.remove()
```

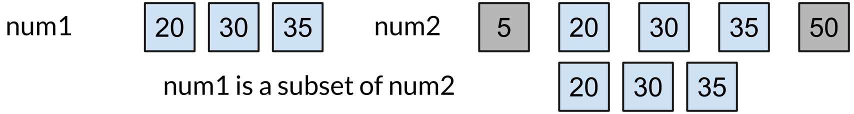
- Tip
  - tips: use the **try-except** structure to catch the error when there is no value to delete and “remove( )” raise error

no return value

Assignment 7-5 : Subset of List

- Write a function `isSubset(numbers1, numbers2)` that takes two lists of numbers. The `isSubset()` function to determine if the first list is a subset of the second list, keeping the sequence.
- The `isSubset()` function checks if all of the elements in the first list are present in the second list, in the same order without the broken sequence(continuous).
  - If they are, then the function returns `True`. If they are not, then the function returns `False`.

return True or False



num1 20 30 35

num2 5 20 30 35 50

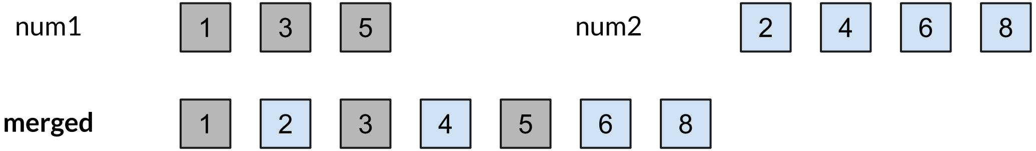
num1 is a subset of num2 20 30 35

- Examples
  - [1, 3, 2] is **not** a subset of [**1, 2, 3**, 4, 5]
  - [1, 3, 2] is a subset of [4, **1, 3, 2**, 5]
  - [1, 4, 5] is **not** a subset of [**1**, 2, 3, **4, 5**]

broken sequence

Assignment 7-6 : Merge two lists

- Write a function `mergeList(num1, num2)` that takes two lists of numbers. The function merge the two lists into a single list, keeping the sorted order. Note: The two lists num1 and num2 are already sorted.



return this

- Requirement
  - Do Not Use **sort( )**
- Return value
  - merged list

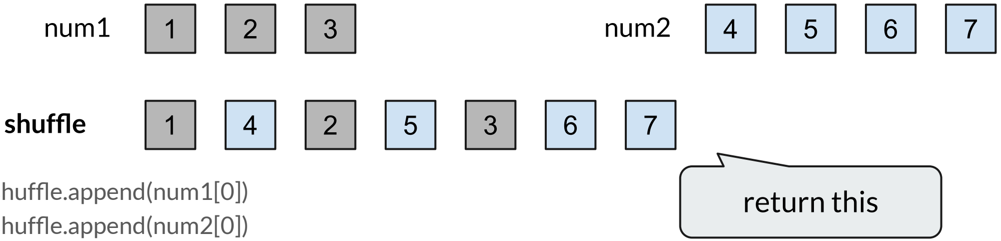
```
Test Data
1 3 5 7
2 4 6 8
Output
1 2 3 4 5 6 7 8
```

```
Test Data
1 2 3 4
5 6 7 8
Output
1 2 3 4 5 6 7 8
```

```
Test Data
5 10 25 75 85
45 55 60
Output
5 10 25 45 55 60 75 85
```

Assignment 7-7 : Shuffle two lists

- Write a function shuffle(num1, num2) that takes two list of numbers and make a new list with shuffling two lists.
    - The shuffle will start with the element in the list num1
    - Take an element from num1 and then num2 until one of the lists is empty
    - If there is no more element in one array, the remainder of the other one will be copied to the last of result array in sequence.



num1 1 2 3

num2 4 5 6 7

shuffle 1 4 2 5 3 6 7

return this

- ○ Hint:
    - shuffle.append(num1[0])
    - shuffle.append(num2[0])
    - and so on…
    - the remainder of num2(the longer list) will be appended to the list shuffle

```
Test data
1 2 3
4 5 6 7
Output
1 4 2 5 3 6 7
```

```
Test Data
1 2 3 4 5
6 7 8
Output
1 6 2 7 3 8 4 5
```

Assignment 7-8 : Substring

- Write a function **`hasARE(words)`** that takes a list of string values and find the words that contains ‘a’, ‘r’ and ‘e’ in sequence
    - ‘a’, ‘r’, and ‘e’ should be **present in sequence(first found ‘a’,then found ‘r’ and then lastly ‘e’)**. Consider only the lowercase letters.
    - Example :
        - words = [‘**are**’, ‘arrow’, ‘**amore**’, ‘**aspire**’, ‘**assertive**’, ‘arrogant’, ‘**bartender**’, ‘**carter**’]
    - Answers = [‘**are**’, ‘**amore**’, ‘**aspire**’, ‘**assertive**’, ‘**bartender**’, ‘**carter**’]
    - Return this answer list
    - Tips:
        - use the string method **`find( )`**

return this

```
test data:  are  arrow  amore  aspire aero
output:   are amore aspire
```

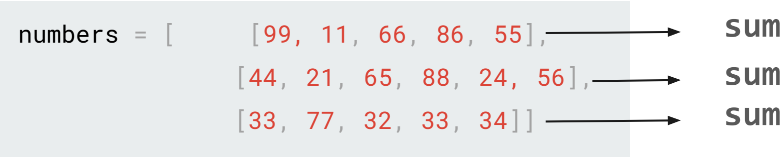
```
test data: assertive arrogant bartender carter racer
Output: assertive bartender carter
```

Assignment 7-9 : two dimensional list

- Write a function **`getMaxElement(numbers)`** that takes two dimensional list(list of list) and then find the greatest number among the entire elements
  - Parameter: list of list[int]
    - The number of elements in a row may be **different**
  - **return value: one integer**
  - In this example, the output is 99

```
numbers = [      [99, 11, 66, 86, 55],
                [44, 21, 65, 88, 24, 56],
                [33, 77, 32, 33, 34]]
```

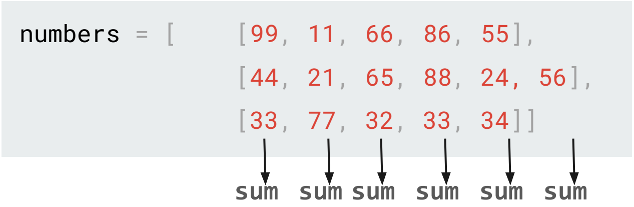
- Write a function **`getSumRows(numbers)`** that takes two dimensional list(list of list) and then returns the list of sums of each row
  - Parameter: list of list[int]
    - The number of elements in a row may be **different**
  - **Return Value: list of integers**
  - Example: [317, 298, 209]



```
numbers = [      [99, 11, 66, 86, 55],
                [44, 21, 65, 88, 24, 56],
                [33, 77, 32, 33, 34]]
```

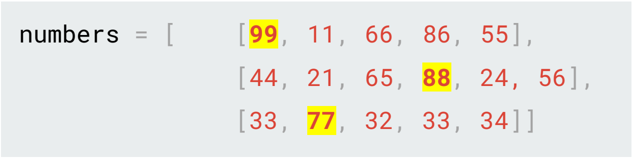
Assignment 7-9 : two dimensional list

- Write a function `getSumCols(numbers)` that takes two dimensional list(list of list) and then returns the list of sums of each column
  - **Parameter: list of list[int]**
    - The number of elements in a row may be **different**
  - **return value: list[int]**
  - Example: [176, 109, 163, 207, 113, 56]



```
numbers = [    [99, 11, 66, 86, 55],
               [44, 21, 65, 88, 24, 56],
               [33, 77, 32, 33, 34]]
```

- Write a function `getMaxElmRow(numbers)` that takes two dimensional list(list of list) and then returns the list of maximum values of each row
  - **Parameter: list of list[int]**
    - The number of elements in a row may be **different**
  - **return value: list[int]**
  - Example: [99, 88, 77]



```
numbers = [    [99, 11, 66, 86, 55],
               [44, 21, 65, 88, 24, 56],
               [33, 77, 32, 33, 34]]
```