# YZV102E - Introduction to Programming for Data Science (Python)
# Lab 3

**09.03.2023**
*Res. Asst. Erhan Biçer (bicer21@itu.edu.tr)*
*Res. Asst. Uğur Önal (onalug@itu.edu.tr)*
*Res. Asst. Yaren Yılmaz (yilmazy20@itu.edu.tr)*

## 1 Exercise 1

In this part, you will complete the following tasks;

1. Get an integer $n1$ from user, it will be the length of the first array. User should give integer greater than 0, repeatedly ask for input if integer is not greater than 0.
2. Create a list named $lst1$ with the length of $n1$ by getting user inputs.
3. Print $lst1$.
4. Get an integer $n2$ from user, it will be the length of the second array. User should give integer greater than 0, repeatedly ask for input if integer is not greater than 0.
5. Create a list named $lst2$ with the length of $n2$ by getting user inputs.
6. Print $lst2$.
7. Calculate the intersection of two lists. **Note:** Intersection should not have the same item twice or more!
8. Print the intersection

## 1.1 Solution of Exercise 1

The solution is given in Code Snippet 1;

Code Snippet 1: Solution of Exercise 1

```python
# step 1
print("Length of the first array:",end="")
while True:
    n1 = input()
    n1 = int(n1)
    if n1 <= 0:
        print("Length should be greater than 0. Please enter again.")
        print("Length of the first array:", end="")
    else:
        break
lst1 = []  # alternative: lst1 = list()
# step 2
for i in range(n1):
    print(f"Enter the {i}. element of the array:",end="")
    new_val = input()
    lst1.append(new_val)
# step 3
print(lst1)
# step 4
print("Length of the second array:",end="")
while True:
    n2 = input()
    n2 = int(n2)
    if n2 <= 0:
        print("Length should be greater than 0. Please enter again.")
        print("Length of the second array:", end="")
    else:
        break
# step 5
lst2 = []
for i in range(n2):
    print(f"Enter the {i}. element of the array:",end="")
    new_val = input()
    lst2.append(new_val)
# step 6
print(lst2)
lst1_set = set(lst1)
lst2_set = set(lst2)
print("Set of list1:")
print(lst1_set)
print("Set of list2:")
print(lst2_set)
# step 7
intersection = list(set(lst1) & set(lst2))

"""
#alternative:
intersection = set()
for i in lst1:
    for j in lst2:
        if i == j:
            intersection.add(i)
"""
# step 8
print("Intersection of two arrays")
print(intersection)

```

## 2 Exercise 2

In this part, you will complete the following tasks;

1. Get an string $str1$ from user.
2. Find the number of digits in the string and print it.
3. Find the number of upper case letters in the string and print it.
4. Find the number of lower case letters in the string and print it.
5. Find the number of vowels in the string and print it.
6. Find the number of consonants in the string and print it.
7. Try these inputs: "Action speaks louder than words!", "YZV 102E/104E is an amazing course!"

## 2.1 Solution of Exercise 2

The solution is given in Code Snippet 3;

Code Snippet 2: Solution of Exercise 2

```
# step 1
str1 = input()

num_digits = 0
num_lowers = 0
num_uppers = 0
num_vowels = 0
num_consonants = 0

for ch in str1:         # iterating over string
        if ch.isdigit():         # check the character is digit or not
                num_digits += 1
        elif ch.isalpha():         # check the character is in alphabet or not
                if ch.isupper():         # check the character is upper or not
                        num_uppers += 1
                else:         # character is lower
                        num_lowers += 1

                if ch.lower() in "aeiou":         # check the character is vowel or not
                        num_vowels += 1
                else:         # character is consonant
                        num_consonants += 1

# printing the results
# step 2
print("num_digits", num_digits)
# step 3
print("num_uppers", num_uppers)
# step 4
print("num_lowers", num_lowers)
# step 5
print("num_vowels", num_vowels)
# step 6
print("num_consonants", num_consonants)

```

# 3 Exercise 3

In this exercise, you will create a dictionary of a reading group.

1. Get an integer from the user, it will be the size of the reading group (dictionary).
2. Create an empty dictionary.
3. Ask *name* and *number of read books* for each member from user and add to the dictionary. Name will be the key and number of read books will be the value.
4. Print the reading group line by line as *name, number of read books*
5. Remove a member from the reading group with given name by the user. If the given name is not in the dictionary, ask repeatedly until user gives a valid name.
6. Print the reading group line by line as *name, number of read books*

## 3.1 Solution of Exercise 3

The solution is given in Code Snippet 3;

Code Snippet 3: Solution of Exercise 3

```python
reading_group = dict()
print("How many people you have in your reading group?" ,end="")
n = input()
n = int(n)  # converting the input to integer from string
for i in range(n):
    print(f"Enter the name of the {i}. member:" ,end="")
    name = input()  # name of the member
    print(f"Enter the number of the books read by {name}:", end="")
    read_books = input()  # number of books read by the member
    read_books = int(read_books) # converting the input to integer from string
    #reading_group[name] = read_books  # add member to the dictionary
    # alternatively:
    reading_group.update({name:read_books})
for key in reading_group.keys():
    print(f"{key},{reading_group[key]}")
print("Your reading group is too crowded, one should leave from this group. Who would leave?",end="")
while True:
    name_removed = input("Which member is going to leave the reading group?")

    if name_removed in reading_group.keys():
        print("User input is valid")
        break
    else:
        print(f"Reading group does not have this member: {name_removed}")
        print("Please enter a valid name")
reading_group.pop(name_leaving)  # remove the record with the given name.
for key in reading_group.keys():
    print(f"{key},{reading_group[key]}")
```