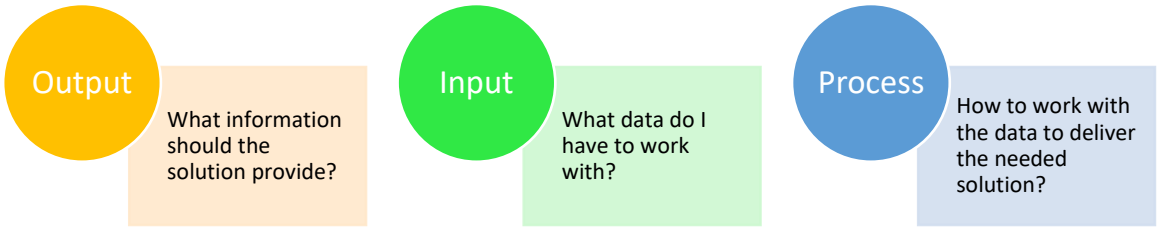
## Instructions:

- You are to solve the following programs in the lab using the problem solving and algorithm methods and share the answer with your lecturer.
- Remember the following rules:
    - Think before you program!
    - A program is a human-readable essay on problem solving that also happens to execute on a computer.
    - The best way to improve your programming and problem skills is to practice!
    - Test your code, often and thoroughly!
    - If it was hard to write, it is probably hard to read.

## Notes:

1. Three steps that a program typically performs:
    - Gather input data:
        - from keyboard
        - from files on disk drives
    - Process the input data
    - Display the results as output:
        - send it to the screen
        - write to a file
2. The moment you read the problem; you should try to answer the following questions:



## Questions

Using the problem-solving techniques, solve the following problems:

1. Calculate area of rectangle.
2. Calculate employee income tax based on the following formula:

   ```
   Tax = 0.25 * (monthly income * 11 – number of kids * 450)
   ```

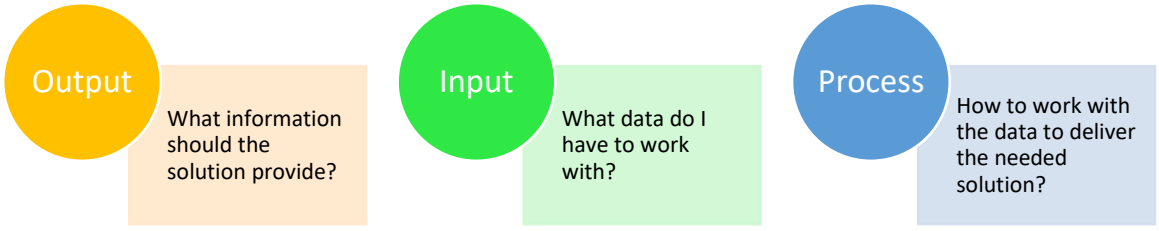
   Your program will display the name of the employee and amount of tax on the screen.
3. Receive an integer from the user, add 5 to it, double it, subtract 7 from it, and display the final number on the screen.
4. Calculate area and circumference of circle.
5. Calculate and print the average of three numbers: 20, 10, and 2.

## Instructions:

- You are to solve the following programs in the lab using the problem solving and algorithm methods and share the answer with your lecturer.

## RECAP (IMPORTANT)

1. Three steps that a program typically performs:
    - Gather input data:
        - from keyboard
        - from files on disk drives
    - Process the input data
    - Display the results as output:
        - send it to the screen
        - write to a file
2. The moment you read the problem; you should try to answer the following questions:




## Questions

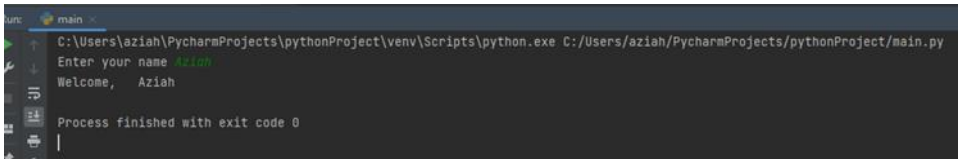
Using the problem-solving techniques, solve the following problems:

1. Using a while loop, asks the user to enter a number, and prints a countdown from the number entered by the user to zero.
2. Check if the number enter is a positive number, negative number or zero.
3. Find the largest numbers between 3 numbers.
4. Check year enter is a leap year or not a leap year.
5. Print odd number(s).
6. Guess if the number that user enter is in a range or not.

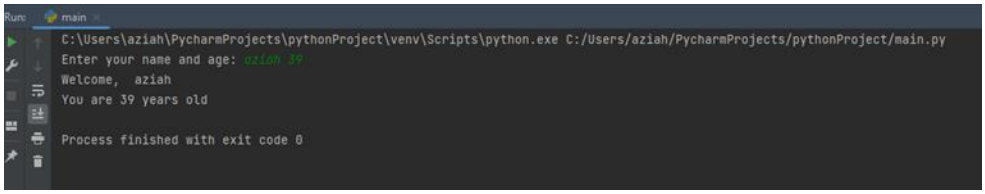
**LAB 3: GETTING STARTED WITH PYTHON**

## Part A: Learn how to use `print` and `input` in Python

1. Print "Hello"
2. Write a program to get an input word from a user. Then, display the given word together with the sentence of "Welcome, " as below:




3. Write a program to get two inputs from user. Then display the input.
   **Output sample:**




## Part B: Test Yourself

Instructions:

- You are to code the following programs in the lab and show the output to your lecturer.
- Write comment to make your programs readable.
- Use descriptive variables in your program (Name of the variables should show their purposes)

1. Accept number from user and display it.
2. Accept 2 numbers from user and display addition of it.
3. Accept 2 input values from user and do arithmetic operation. (+, -, *, /, %).
4. Accept 2 numbers from user and print swapping of those number.
5. Accept 5 subject marks from student and calculate the total and percentage(average) of a student for a semester and display it.
6. Accept Basic from an employee and calculate salary of an employee by considering following things. (Grade_pay is double of Basic. DA is 70% of Basic. TA is RM 200. HRA is 20% of Basic.)(Formula for salary = Grade_pay + DA + TA + HRA).

# LAB 4: CONTROL STRUCTURE IN PYTHON – CONDITIONAL STATEMENT

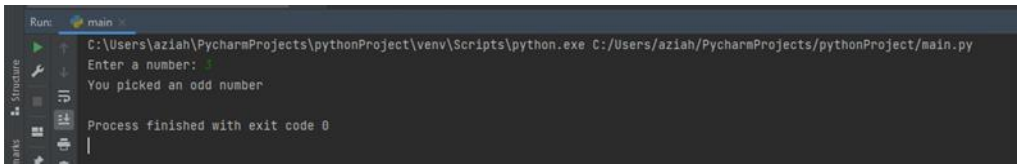
**Instructions:**

- You are to code the following programs in the lab and show the output to your lecturer.
- Write comment to make your programs readable.
- Use descriptive variables in your program (Name of the variables should show their purposes)
- **Note: Design the solution using pseudocode before coding**

**Part A:** Learn how to use the Selection control statements (if, if..else, if..elif..else) in Python

1. Write a program to check if the given number is even or odd.

   **Output sample:**




2. Based on the following table, write a program that determines if your score will pass the exam.

| Score | Output |
|---|---|
| Less than 50 | You failed |
| Other | You passed |

3. Write a PYTHON program to find largest of three numbers
4. Write a program to display profit or loss in trading of an item.

   **Sample input/output:**

   Enter selling price: 20

   Enter buying price:19

   You have profit in trading of this item

**Part B:** Test yourself!

1. Lecturers often provide scores on tests, which are converted to grades the end of the semester. Write a python program to ask for the score input and print the respective grade as follow:

| Score | Grade |
|---|---|
| **Above 80** | A |
| **60 - 79** | B |
| **40 - 59** | C |
| **Below 40** | D |

2. Write a PYTHON program to check entered character is vowel or consonant.
3. Write a PYTHON program to find smallest of three numbers.
4. Write a PYTHON program to check a year is a leap year or not a leap year.
5. Write a Python program to determine whether the speed limit exceeds 110 km per hour. If the speed exceeds 110, then fine = 300, otherwise fine = 0. Display fine.
6. Write a Python program to determine whether the age is above 12 years old. If the age is above 12, then ticket = 20, otherwise ticket = 10. Display ticket.
7. A company insures its drivers in the following cases:
   - If the driver is married.
   - If the driver is unmarried, male and above 30 years of age.
   - If the driver is unmarried, female and above 25 years of age.

   In all the other cases, the driver is not insured.

   If the marital status, gender, and age of the driver are the inputs,

   Write a PYTHON program to determine whether the driver is insured or not

# LAB 5: CONTROL STRUCTURE IN PYTHON – REPETITION STATEMENT

**Instructions:**

- You are to code the following programs in the lab and show the output to your lecturer.
- Write comment to make your programs readable.
- Use descriptive variables in your program (Name of the variables should show their purposes)
- **Note: Design the solution using pseudocode before coding**

**Part A:** Learn how to use the Repetition control statements (for, while) in Python

1. Convert the program design for Question 1, 5 and 6 from previous Lab 2 – Problem Solving Control Structure lab exercise.
2. Write a program that able to print the even number that in the between of 2 to 50.

   **Output sample:**

| | | | |
|---|---|---|---|
| 2 | 4 | 6 | 8 |
| 10 | 12 | 14 | 16 |
| 18 | 20 | 22 | 24 |
| 26 | 28 | 30 | 32 |
| 34 | 36 | 38 | 40 |
| 42 | 44 | 46 | 48 |

**Part B:** Test yourself!

1. Write a Python program to display numbers from 1 to 20. **Note:** use all types of repetitive structures
2. Write a Python program to create a multiplication table of 7.
3. Use a proper iterative structure to write a program that will read **names** and **exam scores** of students. The class **average** is to be calculated and printed at the end of the report. Score can **range from 0 to 100**. Score out of the range is not to be included in the calculations.
4. Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
```

# LAB 6: FUNCTION IN PYTHON

## Instructions:

- You are to code the following programs in the lab and show the output to your lecturer.
- Write comment to make your programs readable.
- Use descriptive variables in your program (Name of the variables should show their purposes)
- **Note: Design the solution using pseudocode before coding**

**Part A:** Function

**Function with variable**

1. Write a program that accepts a variable length of arguments and print the value. Hint: use function

**Return multiple values from a function**

2. Write a program to create function calc() that will accept two variables and calculate the two variables. Hint: Use addition and subtraction.

**Complete the given code below:**

```
def calc(x, y):
    # Write the missing Code

result = calc(40, 10)
print(result)
```

**Function with a default argument**

3. Write a program to create a function named employee() using the following conditions:
    a. Program should accept the employee's name and salary and display both.
    b. If the salary is missing in the function call, then assign default value 9000 to salary.

**Inner function to calculate the addition**

4. Write a Python program to create the following:
    a. Create an outer function that will accept two parameters, y and z.
    b. Create an inner function inside an outer function that will calculate the addition of y and z.
    c. Lastly, the outer function will add 5 into addition and return it

**Assign a different name to function and call the function using the new name.**

5. Based on the example given, assign a new name to the function and call it using the new name.

```
def student(name, age):
    print(name, age)

student("Kelvin", 26)
```

**Built-in Function**

6. Generate a Python list of all the even numbers between 2 to 50.
7. Find the largest number from the given list [4, 28, 97, 56, 16].

**Part B:** Test yourself!

1. Write the following program to find sum of two numbers using a function.

   **Sample input/output:**
   Enter first number: 23
   Enter second number: 7
   Sum of the given two numbers is: 30

2. Write a Python program to read name of student, TP Number and enter his/her all subject marks in list. Compute the total and percentage (Average) of a student. At the end display Name of student, TP Number, Total, Percentage and Grade of that semester by **using function as defined below**.

| Score | Grade |
|---|---|
| **80-100** | A+ |
| **75-79** | A |
| **70-74** | B+ |
| **65-69** | B |
| **60-64** | C+ |
| **55-59** | C |
| **50-54** | C- |
| **40-49** | D |
| **0-39** | F |

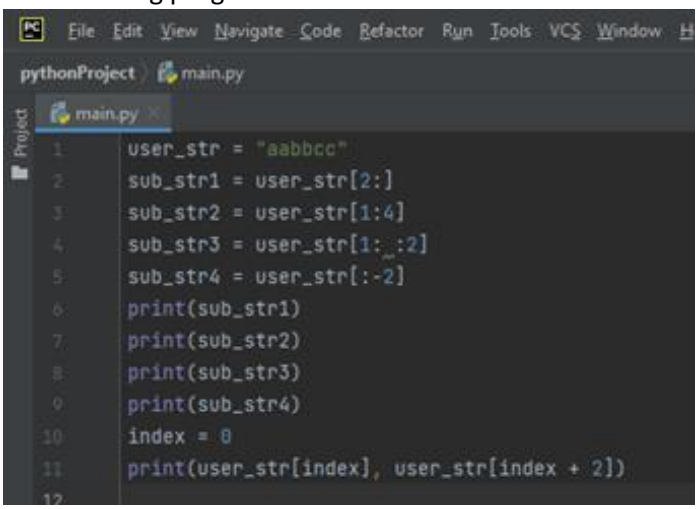
   a) Use **Display function** to print output.
   b) Use **mark function** to accept parameter and return total to Display function.
   c) Use **average function** by passing parameter which is generated in mark function.
   d) Use **grade function** by passing parameter which is generated in average function.

**LAB 7: STRING**

**Part A:** String

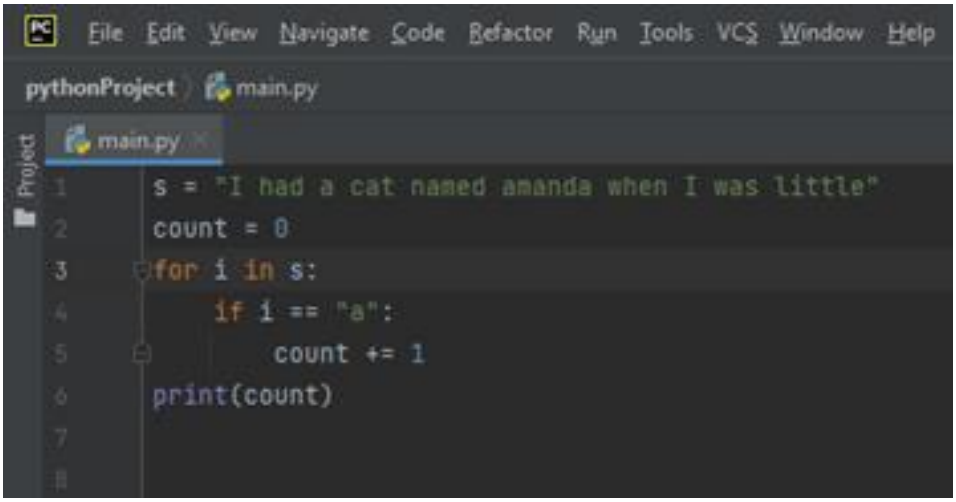
**Slicing Check**

1. Find the output for the following program.



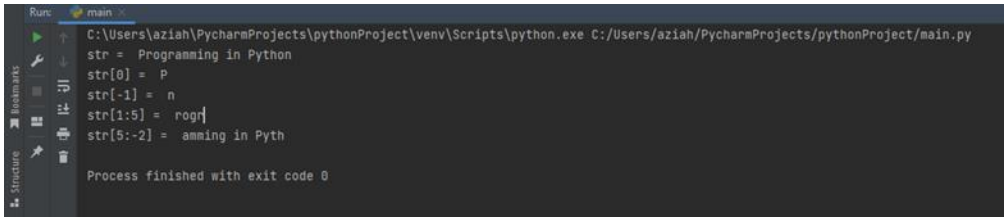

**Comparison Check**

2. In the following program, replace the **for** with a **while** loop.




**Accessing the String**

3. Write a program based on the given output:



**Concatenation of Two or More Strings**

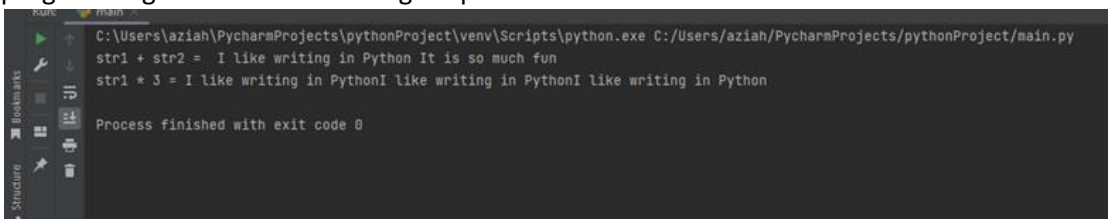
4. We know that writing the following code:

```
print("I like writing in Python")
print("It is so much fun")
```

will produce the following result

```
I like writing in Python
It is so much fun
```

when executed. However, can you manage to produce the given output using the strings? Write a program to generate the following output:




**Part C:** Test yourself!

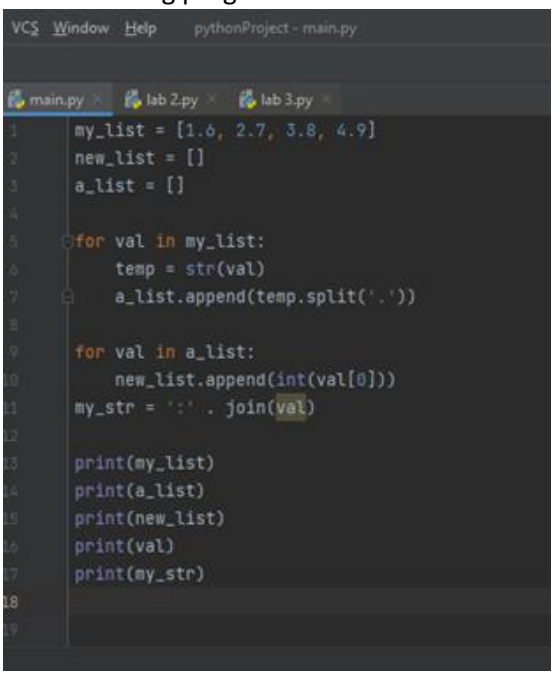
1. Write a Python program to create a new string made of an input of first, middle, and last character.
2. Write a Python program to
   a. Display ' I am a good boy/girl but my parents always scold me'.
   b. Count the length of this string.
   c. Slice this string up to 'Í am a good boy/girl'
   d. Find the word 'parents 'and replace with 'lecturers'
   e. Concatenate 'because I just pretend to be good, HAHAHA'
   f. Display 'good' in upper case
   g. Display "HAHAHA' in lower case
3. Write a Python program to split a given string on hash symbol and display each substring.
   a. Given string = "Magic#Mirror#on#the#wall,#who#is#the#fairest#one#of#all?"
4. Write a Python program to generate below output:
   Given string = "string"

**LAB 8: List**

**Part A:** List

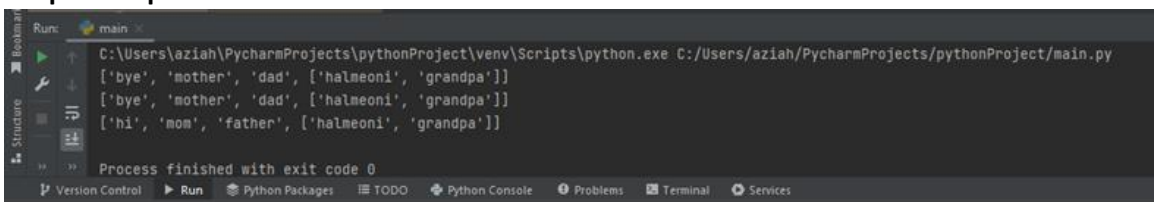
**List and String Check**

1. Generate the output from the following program:




**Mutable List**

2. List is mutable, write a program to generate the given output.
   **Output sample:**




3. Given numList = [1,3,5,5,2]. Write a program that:
   a. Sorts the list.
   b. Adds 4
   c. Remove on duplicate
   d. Inserts 6

e. Print the number of items in the list.

**Part B:** Test yourself!

1. Write a Python code to create a list number using these number = 65, 75, 85, 95, 105 and check number that prompt the user to enter a number to check that number is available or not in list.
2. Write a Python program to shuffle and print a specified list.
   food = [“cookies”, “brownies”, “cake”, “ice cream”, “chocolate”]
3. Write a Python program to get the difference between the two lists.
4. Write a Python program to convert a list of characters into a string.

**LAB 9: Tuple**

**Part A:** Learning Tuple

**Different types of tuple**

1. Generate the output from the following program:

```
# Empty tuple
tuple = ()
print(tuple)

# Tuple with integer value
tuple = (1, 2, 3)
print(tuple)

# Tuple with different datatypes
tuple = (1, "Tuple", 3.7)
print(tuple)

# nested tuple
tuple = ("cat", [100, 98, 65], (1, 2, 3))
print(tuple)
```

**Tuple with One Element**

```
ex1 = ("hello")
print(type(ex1))

ex2 = ("hello",) # creating tuple with 1 element
print(type(ex2))

ex3 = "hello", #creating tuple without parenthesis
print(type(ex3))
```

**Accessing Tuple Element**

1. **Use Index**

```
# Accessing tuple elements using indexing
characters = ("p", "y", "t", "h", "o", "n")

print(characters[0])
print(characters[5])
```

2. **Negative index**

```
# Accessing tuple elements using indexing
characters = ("p", "y", "t", "h", "o", "n")

print(characters[-1])
```

```
print(characters[-3])
```

3. **Slicing**

```
# Accessing tuple elements using slicing
tuple=("p", "y", "t", "h", "o", "n")


print(tuple[1:4])

print(tuple[:-5])

print(tuple[3:])

print(tuple[:])
```

**Part B:** Test yourself!

1. You were given a tuple with elements (2, 4, 5). You want to generate a total value from the tuple elements. Write a Python program to unpack the tuple into several variables and display the total value.
2. You were given a tuple named turtle = (1, “Hello”). Write a python program to
    a. Add on item 5 using +
    b. Add on item “single” using append ().
3. Write a Python program to convert a tuple of characters into a string.
4. Write a Python program to convert a list into tuple.
5. Write a Python program to check if the element exists between a tuple.
6. Given: tuple = (23, 45, 65, 78, 98, 9, 45, 56, 43). Write a Python program to reverse this tuple.

# LAB 10: Set and Dictionary

## Part A: Learn Set

1. **Multiple way to create set**

```
setA = set()

setA = {}

setA = {067069, 076059, 065670, 065590}

vowel_letters = {'a', 'e', 'i', 'o', 'u'}

setA = {'King', 01, -2, 'The Land'}
```

2. **Duplicate Items in a Set**

```
numbers = {1, 2, 3, 3, 1, 5}

print(numbers)
```

3. **Adding Items in a Set**

   Given a set with items as below:

   ```
   numbers = {145, 100, 65, 79}
   ```

   a. Use `add()` to add 94 to the set
   b. Use `update()` to add list into set [23, 56, 78]

4. **Remove item from set**

   Using the same set elements from Part A:3. Use `discard()` to remove 100.

## Part B: Learn Dictionary

1. **Creating a dictionary**

```
learnABC = { "a":"apple","b" :"ball","c":"cat","d":"doll"}

print(learnABC)
```

2. **Valid and invalid dictionary**

   **Which one is the valid or invalid dictionary?**

```
myData = {
  1: "Python",
  (1, 2): 1,
  3: [1, 2, 3]
}
print(myData)
```

```
myData = {
  1: "Python",
  [1, 2]: "2, 3",
}
print(myData)
```

3. **Duplicate in Dictionary**

```
myData = {
    1:"hello",2:22,2:22,3:"hello",3:33
}

print(myData)


myData = {
    1:"hello",2:22,3:22,4:"hello",5:33
}

print(myData)
```

4. **Accessing the dictionary**
   a. Using key

```
languages = {"bm": " Bahasa Melayu", "Eng": "English",
"cn": "cantonese"}

print(languages["bm"])
```

   b. Using `items()` method.

```
print(languages.items())
```

**Part C:** Test yourself!

1. Given a set {1, 2, “Hello”, “Python”}. Write a Python program to iterate over the set.
2. Write a Python program to find element in setA but not in setB.
3. Write a Python program to convert string to set, set to list and set to tuple.
4. Using set, write a Python program to count number of vowels from given strings “Programming with Python”.
5. Write a Python program to add an item (state: Kedah) into the dictionary. Given a dictionary {name: “Amira”, “age”: 35}
6. From question 5 above, write a Python program to display all items in the dictionary using loops.
7. Given a values numbers = {145, 100, 65, 79}, write a Python program to sum all the values in the dictionary.

8. Given a dictionary keys and items myData = { 1:"hello",2:22,2:22,3:"hello",3:33}, write a Python program to check if multiple keys exist in a dictionary.

**LAB 11: File Handling**

**Part A:** File

1. Create a file using mode 'x'
   ```
   file = open("test.txt", "x")
   ```
2. File handling


3. Insert multiple lines into a file
   ```
   file.writelines(["Hello world", "I'm Python", "Don't afraid to use me\n"])
   ```
4. Reads all the lines in file.
   ```
   print(file.readline())
   ```
5. Read from a file line by line
   ```
   file = open('practice.txt', 'w')
   file.writelines("Hi!\nWelcome to the python programming\nLet's continue exploring")
   file = open('practice.txt', 'r')

   for python in file:
       print(python)

   file.close()
   ```
6. Count the lines in text file
   ```
   count = 0
   for line in file:
       count = count + 1

   print('Line Count:', count)
   ```

7. Read length of the text in a file and display string using slice.

```
inp = file.read()

print(len(inp))

print(inp[0:20])
```

8. Read the text file line by line and remove any whitespace using strip()

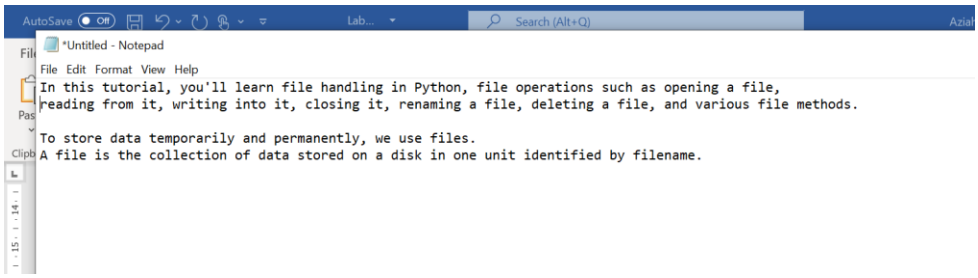
```
while(True):
   line = file.readline()
   if not line:
      break

   print(line.strip())
```

9. Read UTF-8 text files

```
with open('practice.txt', encoding='utf8') as f:
    for line in f:
        print(line.strip())
```

**Part B:** Test yourself!

1. Write a program that opens an output file with the filename "`my.txt`", writes the name of animal, some fruit, and your country to the file on separate lines, then close the file.
2. Write a function in python to read the content from a text file "python.txt" line by line and display the same on screen.
   Create a text file and store the lines as below:




3. Write a function in python to count the number of lines from a text file "`story.txt`" which is not starting with an alphabet "T".
   The file "`story.txt`" should contains any story of yours or you may use the following lines:

> Once upon a time, a farmer had a goose that laid a golden egg every day. The egg provided enough money for the farmer and his wife for their day-to-day needs. The farmer and his wife were happy for a long time. But one day, the farmer got an idea and thought, "Why should I take just one egg a day? Why can't I take all of them at once and make a lot of money?"

4. Write a Python program to remove duplicate lines from a text file. Your program should:
   a. Write a content below into a text file named "`Revision.txt`"

```
Hello this is Python
Good Morning
Hello this is Python
```

```
I love Python
coding is the best!
```

b. Create an empty text file named "`newRevision.txt`". You will write the new content after removing the duplicate lines into this file.
c. Create an empty list to store the unique lines of the file.