Problem Set 1

Question 1 : What data type is each of the following (evaluate where necessary)? 5 5.0 5 > 1 '5' 5 * 2 '5' * 2 '5' + '2' 5 / 2 5 % 2 {5, 2, 1} 5 == 3 Pi (the number)

```
value1 = 5
print(f"{value1} is of type: {type(value1)}")

value2 = 5.0
print(f"{value2} is of type: {type(value2)}")

value3 = 5 > 1
print(f"{value3} is of type: {type(value3)}")

value4 = '5'
print(f"'{value4}' is of type: {type(value4)}")

value5 = 5 * 2
print(f"{value5} is of type: {type(value5)}")

value6 = '5' * 2
print(f"'{value6}' is of type: {type(value6)}")

value7 = '5' + '2'
print(f"'{value7}' is of type: {type(value7)}")

value8 = 5 / 2
print(f"{value8} is of type: {type(value8)}")

value9 = 5 % 2
print(f"{value9} is of type: {type(value9)}")

value10 = {5, 2, 1}
print(f"{value10} is of type: {type(value10)}")

value11 = 5 == 3
print(f"{value11} is of type: {type(value11)}")

import math
value12 = math.pi
print(f"{value12} is of type: {type(value12)}")
```

```
5 is of type: <class 'int'>
5.0 is of type: <class 'float'>
True is of type: <class 'bool'>
'5' is of type: <class 'str'>
```

```
10 is of type: <class 'int'>
'55' is of type: <class 'str'>
'52' is of type: <class 'str'>
2.5 is of type: <class 'float'>
1 is of type: <class 'int'>
{1, 2, 5} is of type: <class 'set'>
False is of type: <class 'bool'>
3.141592653589793 is of type: <class 'float'>
```

Question 2 Write (and evaluate) python expressions that answer these questions:

a. How many letters are there in 'Supercalifragilisticexpialidocious'?

```
word = 'Supercalifragilisticexpialidocious'
letter_count = len(word)
print(f"There are {letter_count} letters in '{word}'")
```

```
There are 34 letters in 'Supercalifragilisticexpialidocious'
```

b. Does 'Supercalifragilisticexpialidocious' contain 'ice' as a substring?

```
word = 'Supercalifragilisticexpialidocious'
substring = 'ice'
contains_substring = substring in word
print(f"'{word}' contains '{substring}': {contains_substring}")
```

```
'Supercalifragilisticexpialidocious' contains 'ice': True
```

c. Which of the following words is the longest: 'Supercalifragilisticexpialidocious', 'Honorificabilitudinitatibus', or 'Bababadalgharaghtakamminarronnkonn'?

```
words = ['Supercalifragilisticexpialidocious',
'Honorificabilitudinitatibus', 'Bababadalgharaghtakamminarronnkonn']
longest_word = max(words, key=len)
print(f"The longest word among the given options is
'{longest_word}'.")
```

```
The longest word among the given options is
'Supercalifragilisticexpialidocious'.
```

d. Which composer comes first in the dictionary: 'Berlioz', 'Borodin', 'Brian', 'Bartok', 'Bellini', 'Buxtehude', 'Bernstein'. Which one comes last?

```
composers = ['Berlioz', 'Borodin', 'Brian', 'Bartok', 'Bellini',
'Buxtehude', 'Bernstein']
first_composer = min(composers)
last_composer = max(composers)
print(f"The first composer in the dictionary is '{first_composer}'.")
print(f"The last composer in the dictionary is '{last_composer}'.")
```

```
The first composer in the dictionary is 'Bartok'.
The last composer in the dictionary is 'Buxtehude'.
```

Question 3 : Implement function triangleArea(a,b,c) that takes as input the lengths of the 3 sides of a triangle and returns the area of the triangle. By Heron's formula, the area of a triangle with side lengths a, b, and c is s(s - a)(s - b)(s - c), where s = (a +b +c)/2.

```
import math

def triangleArea(a, b, c):
    s = (a + b + c) / 2
    area = math.sqrt(s * (s - a) * (s - b) * (s - c))
    return area


result = triangleArea(2, 2, 2)
print(result)
```

```
1.7320508075688772
```

Question 4 Write a program in python to separate odd and even integers in separate arrays. Go to the editor Test Data : Input the number of elements to be stored in the array :5 Input 5 elements in the array : element - 0 : 25 element - 1 : 47 element - 2 : 42 element - 3 : 56 element - 4 : 32

```
num_elements = int(input("Input the number of elements to be stored in the array: "))

even_numbers = []
odd_numbers = []

for i in range(num_elements):
    element = int(input(f"element - {i} : "))
    if element % 2 == 0:
        even_numbers.append(element)
    else:
        odd_numbers.append(element)

print("The Even elements are:")
print(even_numbers)

print("The Odd elements are:")
print(odd_numbers)
```

```
Input the number of elements to be stored in the array: 5
element - 0 : 25
element - 1 : 47
element - 2 : 42
element - 3 : 56
element - 4 : 32
```

```
The Even elements are:
[42, 56, 32]
The Odd elements are:
[25, 47]
```

Question 5

a. Write a function inside(x,y,x1,y1,x2,y2) that returns True or False depending on whether the point (x,y) lies in the rectangle with lower left corner (x1,y1) and upper right corner (x2,y2).

b. Use function inside() from part a. to write an expression that tests whether the point (1,1) lies in both of the following rectangles: one with lower left corner (0.3, 0.5) and upper right corner (1.1, 0.7) and the other with lower left corner (0.5, 0.2) and upper right corner (1.1, 2).

```
def inside(x, y, x1, y1, x2, y2):
    return x1 <= x <= x2 and y1 <= y <= y2

result1 = inside(1, 1, 0, 0, 2, 3)
print(result1)

result2 = inside(-1, -1, 0, 0, 2, 3)
print(result2)

rectangle1 = inside(1, 1, 0.3, 0.5, 1.1, 0.7)
rectangle2 = inside(1, 1, 0.5, 0.2, 1.1, 2)

if rectangle1 and rectangle2:
    print("The point (1,1) lies in both rectangles.")
else:
    print("The point (1,1) does not lie in both rectangles.")
```

```
True
False
The point (1,1) does not lie in both rectangles.
```

Question 6 You can turn a word into pig-Latin using the following two rules (simplified): • If the word starts with a consonant, move that letter to the end and append 'ay'. For example, 'happy' becomes 'appyhay' and 'pencil' becomes 'encilpay'. • If the word starts with a vowel, simply append 'way' to the end of the word. For example, 'enter' becomes 'enterway' and 'other' becomes 'otherway' . For our purposes, there are 5 vowels: a, e, i, o, u (so we count y as a consonant). Write a function pig() that takes a word (i.e., a string) as input and returns its pig-Latin form. Your function should still work if the input word contains upper case characters. Your output should always be lower case however.

```
def pig(word):
    word = word.lower()

    vowels = ['a', 'e', 'i', 'o', 'u']

    if word[0] in vowels:
```

```
        pig_latin_word = word + 'way'
    else:
        index = 0
        for i, letter in enumerate(word):
            if letter in vowels:
                index = i
                break

        pig_latin_word = word[index:] + word[:index] + 'ay'

    return pig_latin_word

print(pig('happy'))
print(pig('Enter'))
```

```
appyhay
enterway
```

Question 7 File bloodtype1.txt records blood-types of patients (A, B, AB, O or OO) at a clinic. Write a function bldcount() that reads the file with name name and reports (i.e., prints) how many patients there are in each bloodtype.

```
def bldcount(filename):
    blood_type_count = {'A': 0, 'B': 0, 'AB': 0, 'O': 0, 'OO': 0}

    try:
        with open(filename, 'r') as file:
            blood_types = file.read().split()

            for blood_type in blood_types:
                if blood_type in blood_type_count:
                    blood_type_count[blood_type] += 1

        for blood_type, count in blood_type_count.items():
            if count == 1:
                print(f'There is one patient of blood type
{blood_type}.')
            elif count > 1:
                print(f'There are {count} patients of blood type
{blood_type}.')
            else:
                print(f'There are no patients of blood type
{blood_type}.')

    except FileNotFoundError:
        print(f"File '{filename}' not found.")

bldcount('bloodtype1.txt')
```

```
There are 15 patients of blood type A.
There is one patient of blood type B.
There are 13 patients of blood type AB.
There are 15 patients of blood type O.
There are no patients of blood type OO.
```

Question 8 Write a function curconv() that takes as input:

1. a currency represented using a string (e.g., 'JPY' for the Japanese Yen or 'EUR' for the Euro)
2. an amount and then converts and returns the amount in US dollars.

```
def curconv(currency, amount):
    conversion_rates = {}

    try:
        with open('currencies.txt', 'r') as file:
            file_content = file.read()

            lines = file_content.strip().split('\n')

            for line in lines:
                parts = line.strip().split()
                if len(parts) == 3:
                    code, rate = parts[0], float(parts[1])
                    conversion_rates[code] = rate

        if currency in conversion_rates:
            usd_amount = amount * conversion_rates[currency]
            return usd_amount
        else:
            return f"Currency code '{currency}' not found in the
database."

    except FileNotFoundError:
        return "File 'currencies.txt' not found."

usd1 = curconv('EUR', 100)
print(usd1)

usd2 = curconv('JPY', 100)
print(usd2)
```

```
122.96544
Currency code 'JPY' not found in the database.
```

Question 9 Each of the following will cause an exception (an error). Identify what type of exception each will cause. Trying to add incompatible variables, as in adding 6 + 'a' Referring to the 12th item of a list that has only 10 items Using a value that is out of range for a function's input, such as calling math.sqrt(-1.0) Using an undeclared variable, such as print(x) when x has

not been defined Trying to open a file that does not exist, such as mistyping the file name or looking in the wrong directory.

1. Trying to add incompatible variables, as in adding 6 + 'a': Because we cannot directly add a string to an integer, trying to do so will result in a TypeError. A list with just 10 items, but referring to item number 12.

2. Referring to the 12th item of a list that has only 10 items: We'll get an IndexError since attempting to reach a list index beyond its range. Mathematical functions like math.sqrt can't handle input values that are outside their range, an example of which is using math.sqrt(-1.0).

3. Using a value that is out of range for a function's input, such as calling math.sqrt(-1.0): The math.sqrt function's constraints do not allow for negative inputs, as the function is not defined for negative values. Therefore, a ValueError will arise if a negative number is entered. Defined variable not used, like printing a value for x that was never declared:

4. Using an undeclared variable, such as print(x) when x has not been defined: In the print statement, the variable x is not defined or declared, leading to a NameError. Looking in the wrong directory or mistyping the file name could result in attempting to open a file that isn't there.

5. Trying to open a file that does not exist, such as mistyping the file name or looking in the wrong directory: An incorrect file path or a typo in the filename will result in a FileNotFoundError being raised by Python, indicating that the specified file cannot be located.

Question 10 Encryption is the process of hiding the meaning of a text by substituting letters in the message with other letters, according to some system. If the process is successful, no one but the intended recipient can understand the encrypted message. Cryptanalysis refers to attempts to undo the encryption, even if some details of the encryption are unknown (for example, if an encrypted message has been intercepted). The first step of cryptanalysis is often to build up a table of letter frequencies in the encrypted text. Assume that the string letters is already defined as 'abcdefghijklmnopqrstuvwxyz'. Write a function called frequencies() that takes a string as its only parameter, and returns a list of integers, showing the number of times each character appears in the text. Your function may ignore any characters that are not in letters.

```
def frequencies(text):
    letters = 'abcdefghijklmnopqrstuvwxyz'
    frequency_list = [0] * 26
    text = text.lower()

    for char in text:
        if char in letters:
            index = letters.index(char)
            frequency_list[index] += 1
```

```
    return frequency_list

result1 = frequencies('The quick red fox got bored and went home.')
print(result1)

result2 = frequencies('apple')
print(result2)
```

```
[1, 1, 1, 3, 5, 1, 1, 2, 1, 0, 1, 0, 1, 2, 4, 0, 1, 2, 0, 3, 1, 0, 1,
1, 0, 0]
[1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]
```