

## Assignment -1

### Python Programming

Assignment Date	29 September 2022
Team ID	PNT2022TMID44156
Student Name	Shafya N
Student Roll Number	724019104021
Maximum Marks	2 Marks

#### Question-1:

Write a Python program to sum of all numerical values (positive integers) embedded in a sentence.

#### Solution:

```
import sys,re

def test(stri):

    print("Input some text and numeric values (<ctrl-d> to exit):")

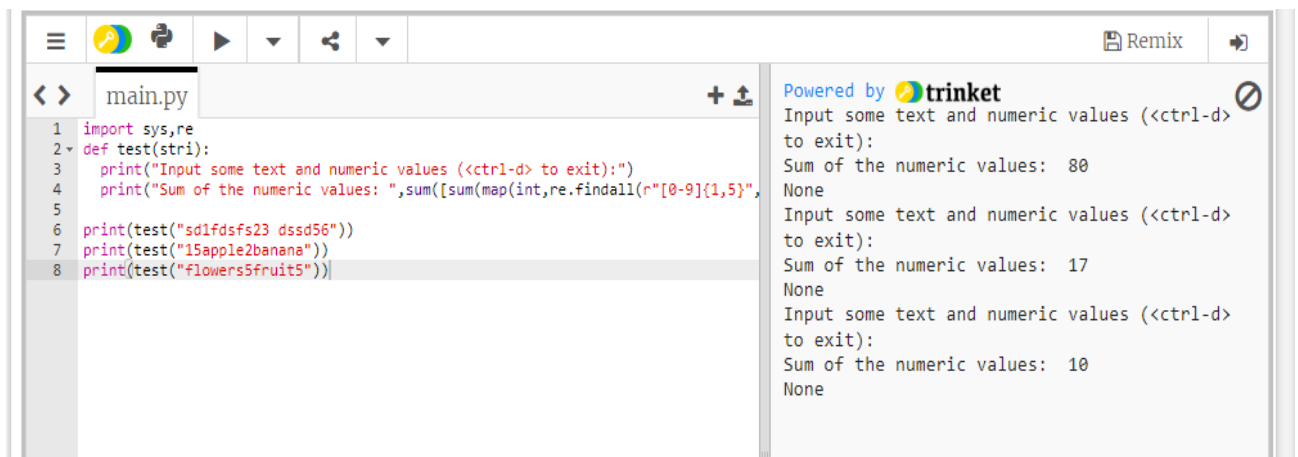
    print("Sum of the numeric values: ",sum([sum(map(int,re.findall(r"[0-9]{1,5}",stri)))]))

print(test("sd1fdsfs23 dssd56"))


print(test("15apple2banana"))

print(test("flowers5fruit5"))
```

#### Output:



```
main.py
1 import sys,re
2 def test(stri):
3     print("Input some text and numeric values (<ctrl-d> to exit):")
4     print("Sum of the numeric values: ",sum([sum(map(int,re.findall(r"[0-9]{1,5}",stri)))]))
5
6 print(test("sd1fdsfs23 dssd56"))
7 print(test("15apple2banana"))
8 print(test("flowers5fruit5"))
```

Powered by  trinket

Input some text and numeric values (<ctrl-d> to exit):  
Sum of the numeric values: 80  
None  
Input some text and numeric values (<ctrl-d> to exit):  
Sum of the numeric values: 17  
None  
Input some text and numeric values (<ctrl-d> to exit):  
Sum of the numeric values: 10  
None

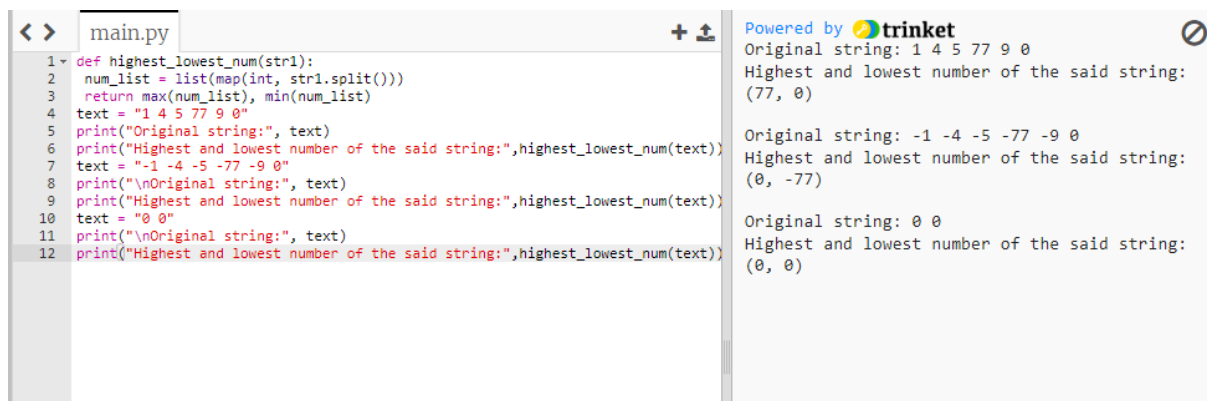
## Question-2:

Write a Python program to find the highest and lowest number from a given string of space separated integers.

### Solution:

```
def highest_lowest_num(str1):
    num_list = list(map(int, str1.split()))
    return max(num_list), min(num_list)
text = "1 4 5 77 9 0"
print("Original string:", text)
print("Highest and lowest number of the said string:",highest_lowest_num(text))
text = "-1 -4 -5 -77 -9 0"
print("\nOriginal string:", text)
print("Highest and lowest number of the said string:",highest_lowest_num(text))
text = "0 0"
print("\nOriginal string:", text)
print("Highest and lowest number of the said string:",highest_lowest_num(text))
```

### Output:

The image shows a screenshot of a Python code editor. On the left, the code is written in a file named 'main.py'. The code defines a function 'highest\_lowest\_num' that takes a string 'str1' and returns a tuple of the maximum and minimum values from the list of integers created by splitting the string. It then tests this function with three different input strings: '1 4 5 77 9 0', '-1 -4 -5 -77 -9 0', and '0 0'. On the right, the output of the program is displayed. It shows the original string for each test case followed by the highest and lowest numbers found. For the first string, the output is (77, 0). For the second, it is (0, -77). For the third, it is (0, 0).

```
< > main.py + ↕
1 def highest_lowest_num(str1):
2     num_list = list(map(int, str1.split()))
3     return max(num_list), min(num_list)
4 text = "1 4 5 77 9 0"
5 print("Original string:", text)
6 print("Highest and lowest number of the said string:",highest_lowest_num(text))
7 text = "-1 -4 -5 -77 -9 0"
8 print("\nOriginal string:", text)
9 print("Highest and lowest number of the said string:",highest_lowest_num(text))
10 text = "0 0"
11 print("\nOriginal string:", text)
12 print("Highest and lowest number of the said string:",highest_lowest_num(text))

Powered by trinket
Original string: 1 4 5 77 9 0
Highest and lowest number of the said string:
(77, 0)

Original string: -1 -4 -5 -77 -9 0
Highest and lowest number of the said string:
(0, -77)

Original string: 0 0
Highest and lowest number of the said string:
(0, 0)
```

## Question-3

Write a Python program to calculate the maximum profit from selling and buying values of stock. An array of numbers represent the stock prices in chronological order.

### Solution:

```
def buy_and_sell(stock_price):
    max_profit_val, current_max_val = 0, 0
```

```

        for price in reversed(stock_price):
            current_max_val = max(current_max_val, price)
            potential_profit = current_max_val - price
            max_profit_val = max(potential_profit, max_profit_val)

    return max_profit_val

print(buy_and_sell([8, 10, 7, 5, 7, 15]))
print(buy_and_sell([1, 2, 8, 1]))
print(buy_and_sell([]))

```

### Output:

The screenshot shows a Python code editor with a file named 'main.py'. The code defines a function 'buy\_and\_sell' that takes a list of stock prices and returns the maximum profit. The function iterates through the prices in reverse order, keeping track of the current maximum value and the potential profit. The code then prints the results of the function for three different input lists: [8, 10, 7, 5, 7, 15], [1, 2, 8, 1], and an empty list. The output on the right shows the results: 10, 7, and 0.

```

1 def buy_and_sell(stock_price):
2     max_profit_val, current_max_val = 0, 0
3     for price in reversed(stock_price):
4         current_max_val = max(current_max_val, price)
5         potential_profit = current_max_val - price
6         max_profit_val = max(potential_profit, max_profit_val)
7
8     return max_profit_val
9
10 print(buy_and_sell([8, 10, 7, 5, 7, 15]))
11 print(buy_and_sell([1, 2, 8, 1]))
12 print(buy_and_sell([]))
13

```

Powered by trinket

10  
7  
0

### Question-4

Write a Python program to test whether a given number is symmetrical or not. A number is symmetrical when it is equal of its reverse.

### Solution:

```

def is_symmetrical_num(n):
    return str(n) == str(n)[::-1]

print(is_symmetrical_num(121))
print(is_symmetrical_num(0))
print(is_symmetrical_num(122))
print(is_symmetrical_num(990099))

```

## Output:



The screenshot shows a Trinket.io code editor with a file named 'main.py'. The code defines a function 'is\_symmetrical\_num(n)' that checks if a string is a palindrome. It then prints the results of this function for the inputs '121', '0', '122', and '990099'. The output on the right shows 'True', 'True', 'False', and 'True' respectively.

```
1 def is_symmetrical_num(n):
2     return str(n) == str(n)[::-1]
3 print(is_symmetrical_num(121))
4 print(is_symmetrical_num(0))
5 print(is_symmetrical_num(122))
6 print(is_symmetrical_num(990099))
7
```

Powered by  trinket

True  
True  
False  
True

## Question-5

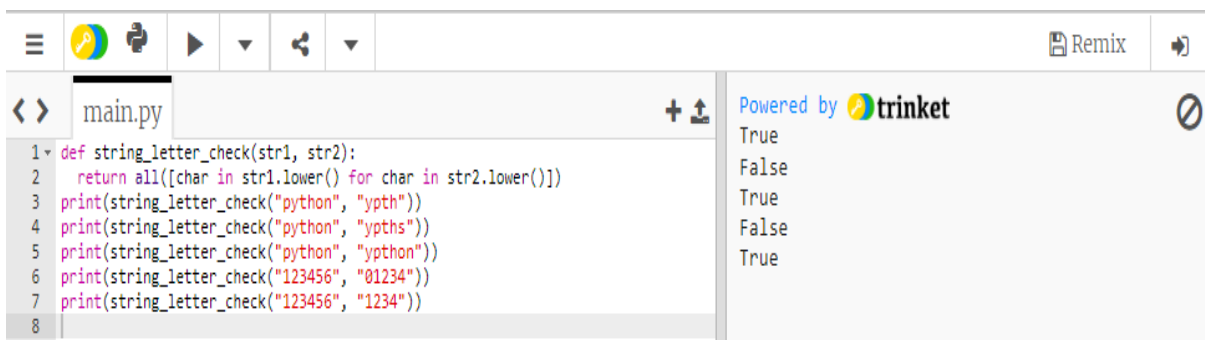
Write a Python program that accept two strings and test if the letters in the second string are present in the first string.

### Solution:

```
def string_letter_check(str1, str2):
    return all([char in str1.lower() for char in str2.lower()])


print(string_letter_check("python", "ypth"))
print(string_letter_check("python", "ypths"))
print(string_letter_check("python", "ypthon"))
print(string_letter_check("123456", "01234"))
print(string_letter_check("123456", "1234"))
```

## Output:



The screenshot shows a Trinket.io code editor with a file named 'main.py'. The code defines a function 'string\_letter\_check(str1, str2)' that checks if all characters in the second string are present in the first string. It then prints the results of this function for the inputs ('python', 'ypth'), ('python', 'ypths'), ('python', 'ypthon'), ('123456', '01234'), and ('123456', '1234'). The output on the right shows 'True', 'False', 'True', 'False', and 'True' respectively.

```
1 def string_letter_check(str1, str2):
2     return all([char in str1.lower() for char in str2.lower()])
3 print(string_letter_check("python", "ypth"))
4 print(string_letter_check("python", "ypths"))
5 print(string_letter_check("python", "ypthon"))
6 print(string_letter_check("123456", "01234"))
7 print(string_letter_check("123456", "1234"))
8
```

Powered by  trinket

True  
False  
True  
False  
True