

TT B I <> ⌂ ⌂ “ “ ⌂ ⌂ – ψ ☺ ⌂

#Python Assignment 2

Python Assignment 2

```
# Task 1
def count_vowels_consonants(s: str):
    vowels = set("aeiou")
    v = c = 0
    for ch in s.lower():
        if ch.isalpha():
            if ch in vowels:
                v += 1
            else:
                c += 1
    return v, c
```

```
# Demo
text = input("Enter text: ")
v, c = count_vowels_consonants(text)
print("Vowels:", v, "Consonants:", c)
```

```
→ Enter text: intelligent
Vowels: 4 Consonants: 7
```

```
# Task 2
def min_max_tuple(t: tuple):
    if not t:
        raise ValueError("Empty tuple has no min/max.")
    return min(t), max(t)

# Demo
t = tuple(map(int, input("Enter numbers (space-separated): ").split()))
print("Min, Max:", min_max_tuple(t))
```

```
→ Enter numbers (space-separated): 76 66 7877 567 77676
Min, Max: (66, 77676)
```

```
# Task 3
t = tuple(input("Enter tuple elements (space-separated): ").split())
first, second, *rest = t
print("First:", first)
print("Second:", second)
print("Rest:", rest)
```

→ Enter tuple elements (space-separated): orange apple mango banana grape pap
First: orange
Second: apple
Rest: ['mango', 'banana', 'grape', 'papaya']

```
# Task 4
import re
from collections import Counter
```

```
def word_frequency(text: str) -> dict:
    words = re.findall(r"[A-Za-z0-9]+", text.lower())
    return dict(Counter(words))
```

```
# Demo
s = input("Enter a sentence: ")
freq = word_frequency(s)
print(freq)
```

→ Enter a sentence: I am an Indian
{'i': 1, 'am': 1, 'an': 1, 'indian': 1}

```
# Task 5
from collections import Counter
```

```
def find_duplicates(items):
    counts = Counter(items)
    return [x for x, cnt in counts.items() if cnt > 1]
```

```
# Demo
lst = input("Enter list items (space-separated): ").split()
print("Duplicates:", find_duplicates(lst))
```

→ Enter list items (space-separated): orange apple grape orange banana papaya
Duplicates: ['orange', 'apple', 'grape']

```
# Task 6
nums = list(map(int, input("Enter integers (space-separated): ").split()))
evens = list(filter(lambda x: x % 2 == 0, nums))
print("Evens:", evens)
```

→ Enter integers (space-separated): 43 66 78 91 56
Evens: [66, 78, 56]

```
# ---- Save this as: mymath.py ----
```

```
def add(a, b):
    return a + b
```

```
def sub(a, b):
    return a - b
```

```
def mul(a, b):
    return a * b
```

```
# Write the content of the mymath.py cell to a file
```

```
with open("mymath.py", "w") as f:
```

```
    f.write(""""
```

```
def add(a, b):
    return a + b
```

```
def sub(a, b):
    return a - b
```

```
def mul(a, b):
    return a * b
"""")
```

```
# ---- Usage example (same folder), e.g., main.py ----
```

```
import mymath
```

```
print("add(3, 4) =", mymath.add(3, 4))
print("sub(10, 6) =", mymath.sub(10, 6))
print("mul(7, 5) =", mymath.mul(7, 5))
```

→ add(3, 4) = 7
sub(10, 6) = 4
mul(7, 5) = 35

```
# Task 8
def sort_by_second(tuples_list):
    return sorted(tuples_list, key=lambda x: x[1])
```

```
# Demo
data = [(1, 3), (2, 2), (3, 1), (4, 2)]
print("Sorted:", sort_by_second(data))
```

→ Sorted: [(3, 1), (2, 2), (4, 2), (1, 3)]

```
# Task 9
def symmetric_difference(a, b):
    return set(a) ^ set(b) # elements in either set, but not both
```

```
# Demo
A = set(input("Set A elements (space-separated): ").split())
B = set(input("Set B elements (space-separated): ").split())
print("Symmetric difference:", symmetric_difference(A, B))
```

→ Set A elements (space-separated): 7678 5786 7556 8788
Set B elements (space-separated): 65767 878 67 8787
Symmetric difference: {'67', '8787', '8788', '65767', '878', '7556', '7678'}

```
# Task 10
def primes_upto(n: int):
    if n < 2:
        return []
    sieve = [True] * (n + 1)
    sieve[0] = sieve[1] = False
    p = 2
    while p * p <= n:
        if sieve[p]:
            for m in range(p * p, n + 1, p):
                sieve[m] = False
        p += 1
    return [i for i, is_prime in enumerate(sieve) if is_prime]
```

```
# Demo
n = int(input("Enter n: "))
print("Primes up to n:", primes_upto(n))
```

→ Enter n: 99
Primes up to n: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53]

By, Utkarsh Anand