Q1. Function to flatten a list and return the product of all numeric values

Requirements covered

- Flatten nested lists, tuples, sets, and dictionaries
- Extract **numeric keys and numeric values** from dictionaries
- Ignore non-numeric values (strings, booleans)
- Return the **product** of all numbers

Program

```
from functools import reduce

def flatten_and_product(lst):
    numbers = []

    def extract(item):
        if isinstance(item, (int, float)) and not isinstance(item,
bool):
            numbers.append(item)

        elif isinstance(item, (list, tuple, set)):
            for i in item:
                extract(i)

        elif isinstance(item, dict):
            for k, v in item.items():
                extract(k)
                extract(v)

    extract(lst)

    return reduce(lambda x, y: x * y, numbers)
```

```
list1 = [
    1, 2, 3, 4,
    [44, 55, 66, True],
    False,
    (34, 56, 78, 89, 34),
    {1, 2, 3, 3, 2, 1},
    {1: 34, "key2": [55, 67, 78, 89], 4: (45, 22, 61, 34)},
    [56, 'data science'],
    'Machine Learning'
]

result = flatten_and_product(list1)
print(result)
```

Q2. Python program to encrypt a message

Encryption rules

- Convert input to **lowercase**
- Reverse the alphabet
    - `a → z`, `b → y`, `c → x`, …
- Replace **whitespace** with `$`
- Keep **punctuation marks unchanged**

Program

```
def encrypt_message(sentence):
    sentence = sentence.lower()
    encrypted = ""

    for ch in sentence:
        if ch.isalpha():
            encrypted += chr(122 - (ord(ch) - 97))
        elif ch == " ":
            encrypted += "$"
        else:
            encrypted += ch

    return encrypted


input_sentence = "I want to become a Data Scientist."
output = encrypt_message(input_sentence)
print(output)
```

Encrypted Output

```
r$dzmg$gl$yvxlnv$z$wzgz$hxrvmgrhg.
```