06.02.2023 Assignment

March 24, 2023

Q1. Create a function which will take a list as an argument and return the product of all the numbers after creating a flat list. Use the below-given list as an argument for your function. 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’] Note: you must extract numeric keys and values of the dictionary also.

[1]:

```python
def product_of_numbers(lst):
    flat_list = []
    for item in lst:
        if isinstance(item, (int, float)):
            flat_list.append(item)
        elif isinstance(item, list):
            flat_list.extend([i for i in item if isinstance(i, (int, float))])
        elif isinstance(item, tuple):
            flat_list.extend([i for i in item if isinstance(i, (int, float))])
        elif isinstance(item, set):
            flat_list.extend([i for i in item if isinstance(i, (int, float))])
        elif isinstance(item, dict):
            flat_list.extend([i for i in item.values() if isinstance(i, (int,␣
↪float))])
            flat_list.extend([i for j in item.items() for i in j if␣
↪isinstance(i, (int, float))])
    product = 1
    for num in flat_list:
        product *= num
    return product
```

[2]:

```python
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 = product_of_numbers(list1)

print(result)
```

```
0
```

Q2. Write a python program for encrypting a message sent to you by your friend. The logic of encryption should be such that, for a the output should be z. For b, the output should be y.

For c, the output should be x respectively. Also, the whitespace should be replaced with a dollar sign. Keep the punctuation marks unchanged. Input Sentence: I want to become a Data Scientist. Encrypt the above input sentence using the program you just created. Note: Convert the given input sentence into lowercase before encrypting. The final output should be lowercase.

[3]:

```
def encrypt(message):
    encrypted = ""
    for char in message.lower():
        if char.isalpha():
            encrypted += chr(219 - ord(char))
        elif char == " ":
            encrypted += "$"
        else:
            encrypted += char
    return encrypted

input_sentence = "I want to become a Data Scientist."
encrypted_sentence = encrypt(input_sentence)

print(encrypted_sentence)
```

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

[4]:

```
def encrypt(message):
    encrypted = ""
    for char in message.lower():
        if char.isalpha():
            encrypted += chr(219 - ord(char))
        elif char == "$":
            encrypted += " "
        else:
            encrypted += char
    return encrypted

input_sentence = "r$dzmg$gl$yvxlnv$z$wzgz$hxrvmgrhg."
encrypted_sentence = encrypt(input_sentence)

print(encrypted_sentence)
```

```
i want to become a data scientist.
```

[ ]:

```

```