Python basic tasks

PALINDROME FUNCTION:

```python
# Sample Palindrome function

def looping(data):
    lis = []
    reversedStr = ""
    for i in data:
        if type(data) == str:
            lis.append(i)
        else:
            reversedStr += i
    if len(reversedStr) > 0:
        return reversedStr
    elif len(lis) > 0:
        lis.reverse()
        return lis


def palindrome(data):
    lis = looping(data)
    reversed = looping(lis)
    if data == reversed:
        print("given string is palindrome")
    else:
        print("given string is not palindrome")


palindrome("madam")
```

FACTORIAL FUNCTION:

```python
#  Factorial method without recursion

def normalFactorial(num):
    mulStr = 1
    for i in range(num):
        mulStr = mulStr * (i+1)
    return mulStr

nonRecResult =  normalFactorial(7)

print("non recursion result--->", nonRecResult)
```

```python
#  Factorial method with recursion

def recursionFactorial(num):
    if num == 0:
        return 1
    else:
        return num * recursionFactorial(num-1)

recResult =  recursionFactorial(7)
print("recursion result ------>", recResult)
```

FIBONACCI SERIES

```python
#  fibonacci series without recursion

def nonRecursionFibonacci(length):
    defaultDigi = [0, 1]
    for i in range(length):
        defaultDigi.append(defaultDigi[-1] + defaultDigi[-2])
    return defaultDigi

nonRecResult = nonRecursionFibonacci(10)
print(nonRecResult)

#  fibonacci series with recursion
lis = [0, 1]
def recursionFibonacci(length):
    if len(lis) != length:
        lis.append(lis[-1] + lis[-2])
        recursionFibonacci(length)



recursionFibonacci(10)
print("rec result", lis)
```

LIST & DICTIONARY TASKS

```python
lis  =[
    "apple",
    "orange",
    "grapes",
    24, 65, True,
    "Ram",
    "sundar",
    [
        "David",
        "Gobi",
        "Sudakar",
        [
            "Ipads",
            "Android",
            "iPhone",
            "samsung",
            "Vivo",
            "add oppo here", # needs to be changed in this line
            {
                "name": "Android",
                "price": 5000,
                "color": "black",
                "ram": "2GB",
                "storage": "1GB",
                "companies": ["samsung", "vivo", "replace new company name here", "oppo"], # needs to be updated in this line
                "processor": "arm",
                "screen": 5.5,
                "weight": 50,
            }
        ],
        "deeplink", # change this deeplink to firefox
        "chrome",
        "electron",
        "react",
        "vue"
    ],
    # copy the third level list and place it here in this line
    ]

# tasks
# 1. find the length of the overall list.
# 2. get the third level list and print the even number keys on the dictionary.
# 3. take out all the data from the list and print seperately using recursion.
```