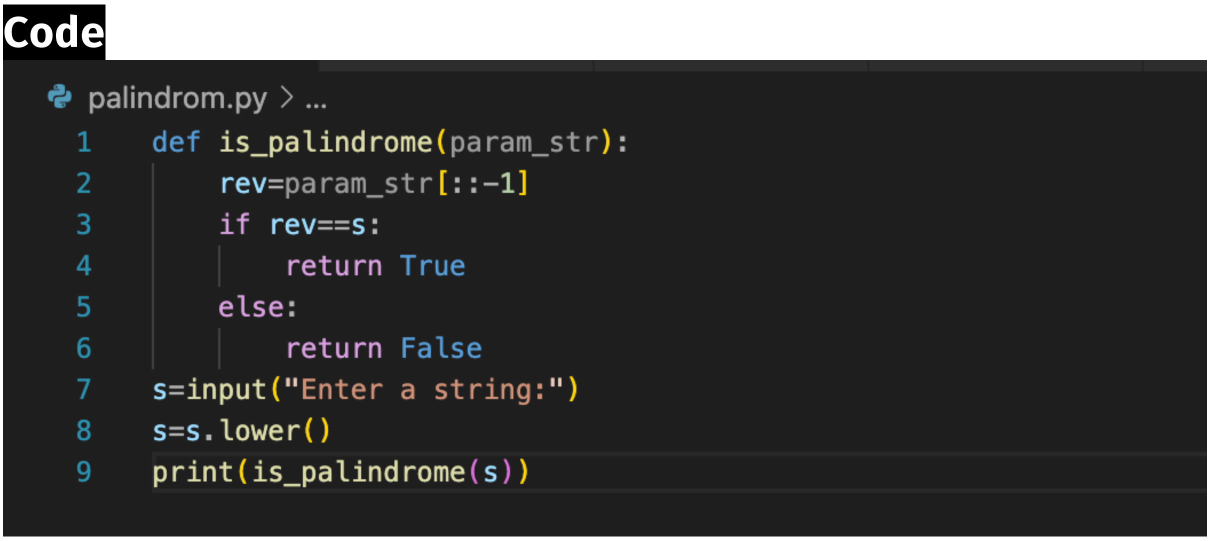
PYTHON
PROGRAMMING CHALLENGES
4.1.
check palindrome

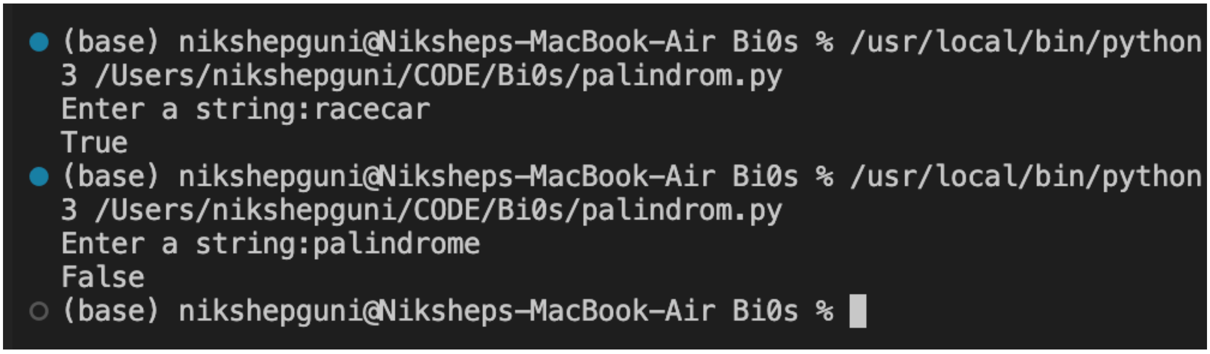
Problem: Write a function is_palindrome(s) that takes a string and returns True if the string is a palindrome (reads the same forwards and backwards), and False otherwise..

Code



```python
def is_palindrome(param_str):
    rev=param_str[::-1]
    if rev==s:
        return True
    else:
        return False
s=input("Enter a string:")
s=s.lower()
print(is_palindrome(s))
```

Output

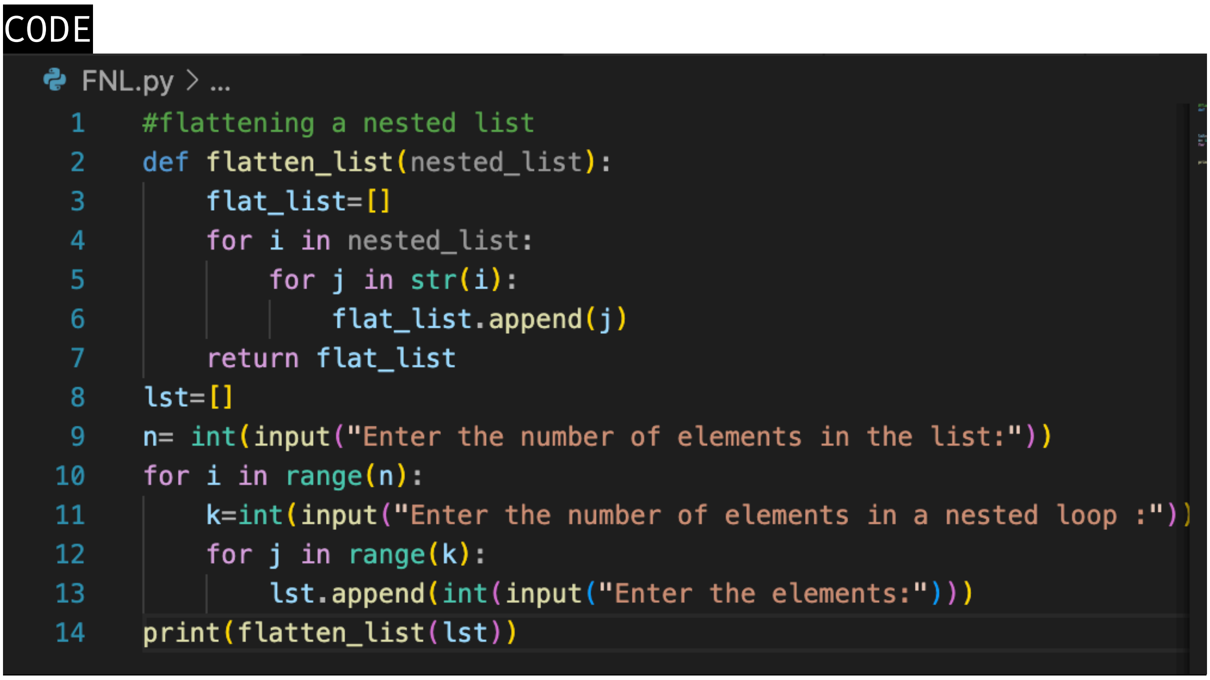


```
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/palindrom.py
  Enter a string:racecar
  True
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/palindrom.py
  Enter a string:palindrome
  False
○ (base) nikshepguni@Niksheps-MacBook-Air Bi0s % █
```

Flatten Nested List

Problem: Write a function flatten_list(nested_list) that takes a list which may contain nested lists, and returns a flat list with all the elements.

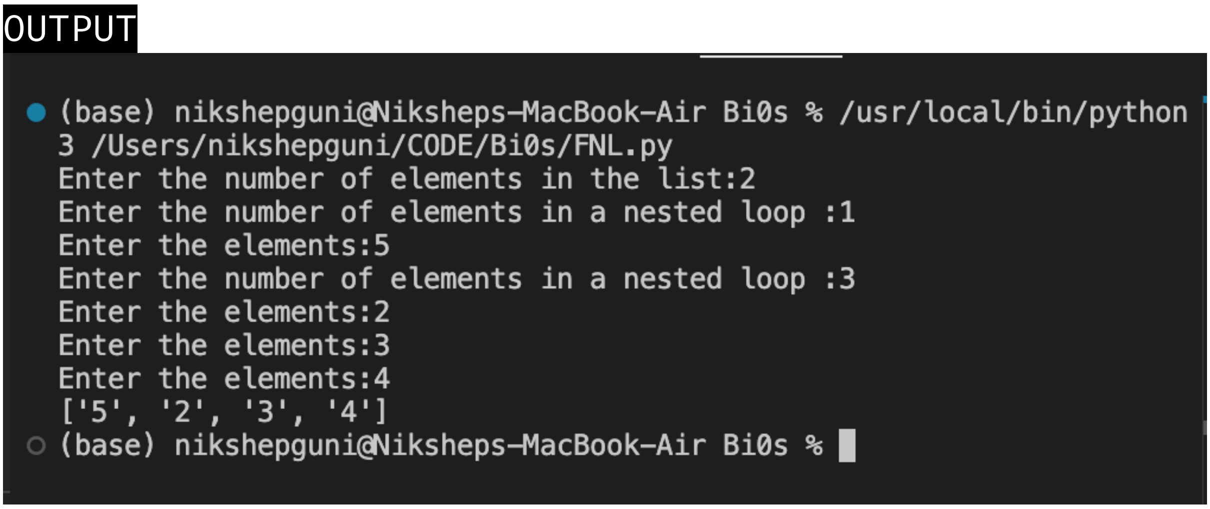
CODE



FNL.py > ...

```python
#flattening a nested list
def flatten_list(nested_list):
    flat_list=[]
    for i in nested_list:
        for j in str(i):
            flat_list.append(j)
    return flat_list
lst=[]
n= int(input("Enter the number of elements in the list:"))
for i in range(n):
    k=int(input("Enter the number of elements in a nested loop :"))
    for j in range(k):
        lst.append(int(input("Enter the elements:")))
print(flatten_list(lst))
```

OUTPUT

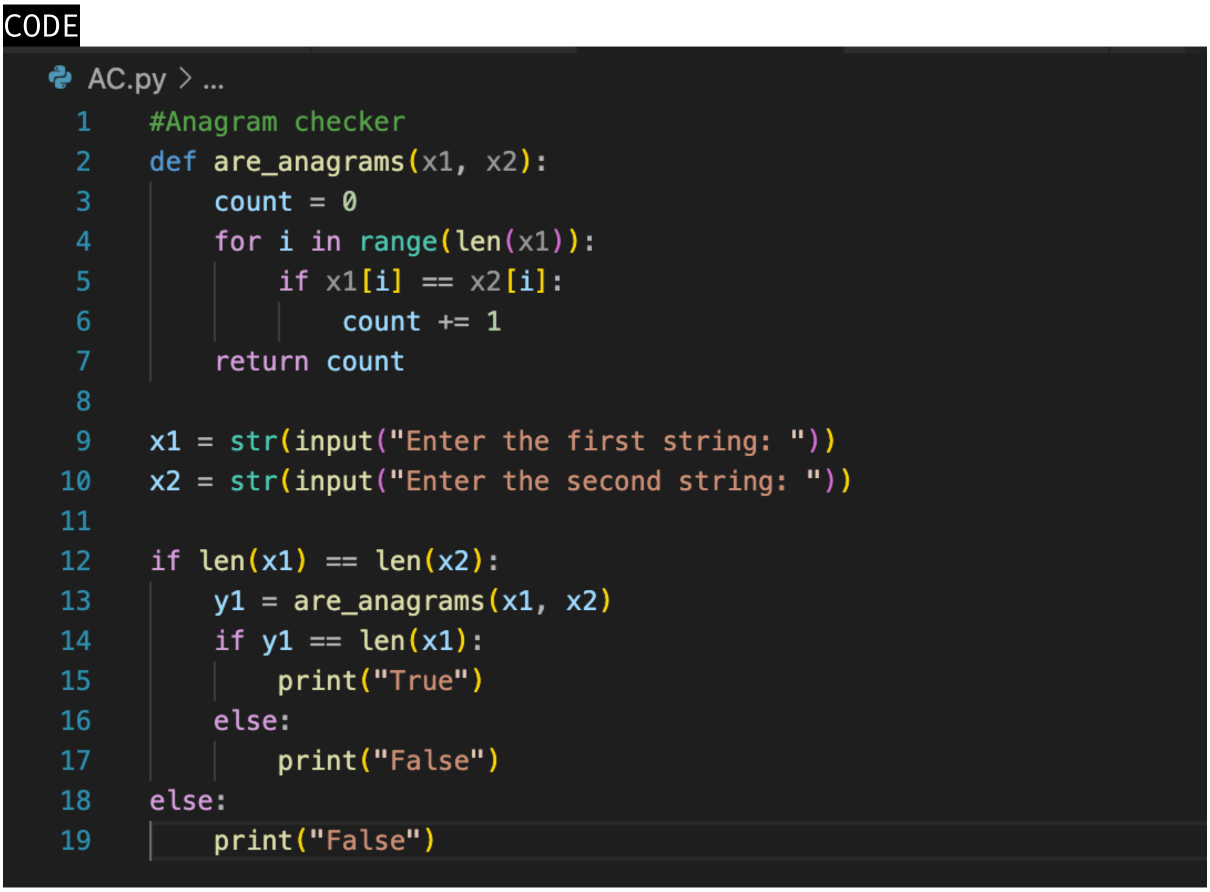


```
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/FNL.py
  Enter the number of elements in the list:2
  Enter the number of elements in a nested loop :1
  Enter the elements:5
  Enter the number of elements in a nested loop :3
  Enter the elements:2
  Enter the elements:3
  Enter the elements:4
  ['5', '2', '3', '4']
○ (base) nikshepguni@Niksheps-MacBook-Air Bi0s % █
```

ANAGRAM CHECKER

Problem: Write a function are_anagrams(s1, s2) that takes two strings and returns True if they are anagrams of each other, and False otherwise.

CODE



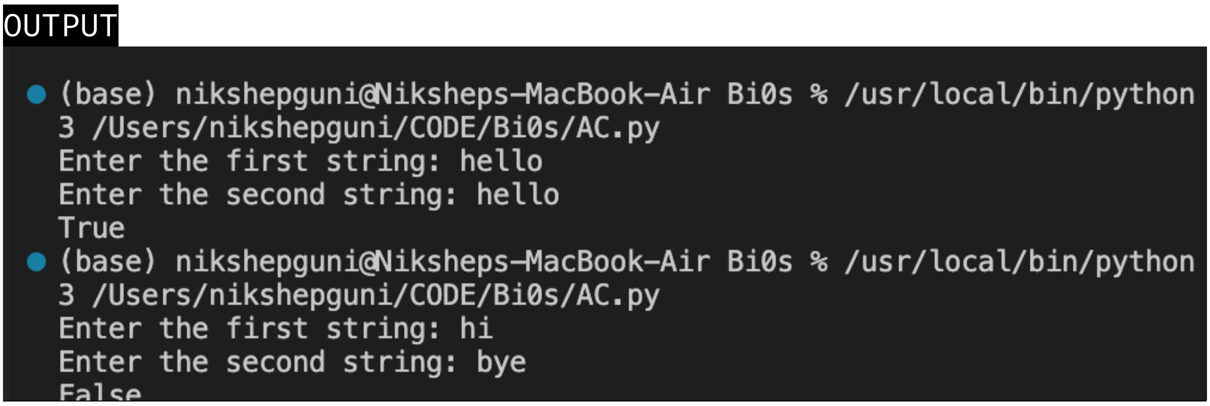
AC.py > ...

```python
#Anagram checker
def are_anagrams(x1, x2):
    count = 0
    for i in range(len(x1)):
        if x1[i] == x2[i]:
            count += 1
    return count

x1 = str(input("Enter the first string: "))
x2 = str(input("Enter the second string: "))

if len(x1) == len(x2):
    y1 = are_anagrams(x1, x2)
    if y1 == len(x1):
        print("True")
    else:
        print("False")
else:
    print("False")
```

OUTPUT

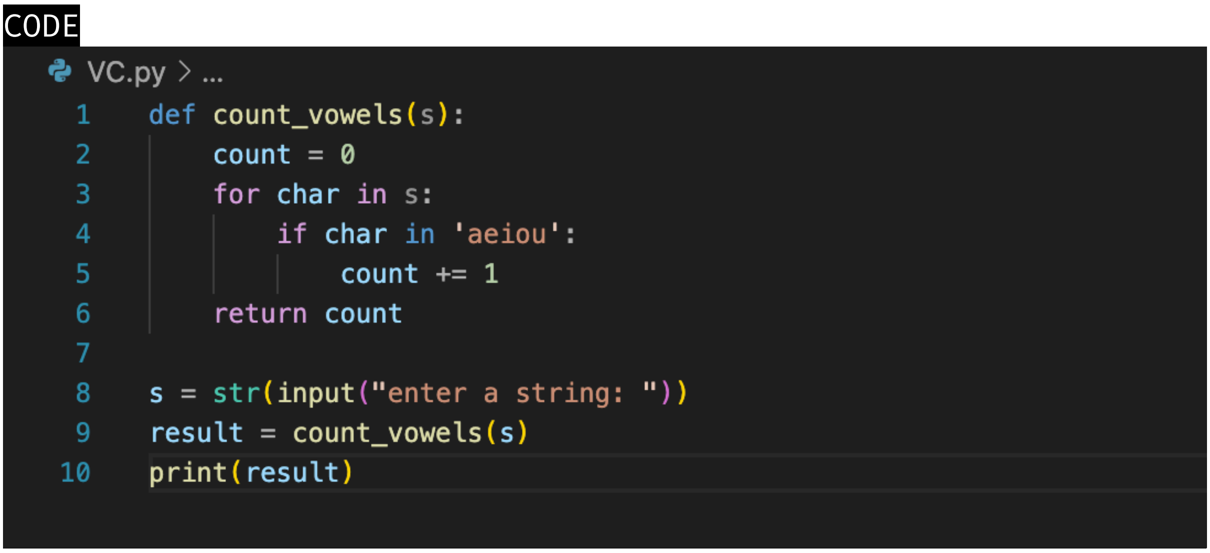


```
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/AC.py
  Enter the first string: hello
  Enter the second string: hello
  True
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/AC.py
  Enter the first string: hi
  Enter the second string: bye
  False
```

VOWEL COUNTER

Problem: Write a function count_vowels(s) that takes a string and returns the number of vowels (a, e, i, o, u) in the string.

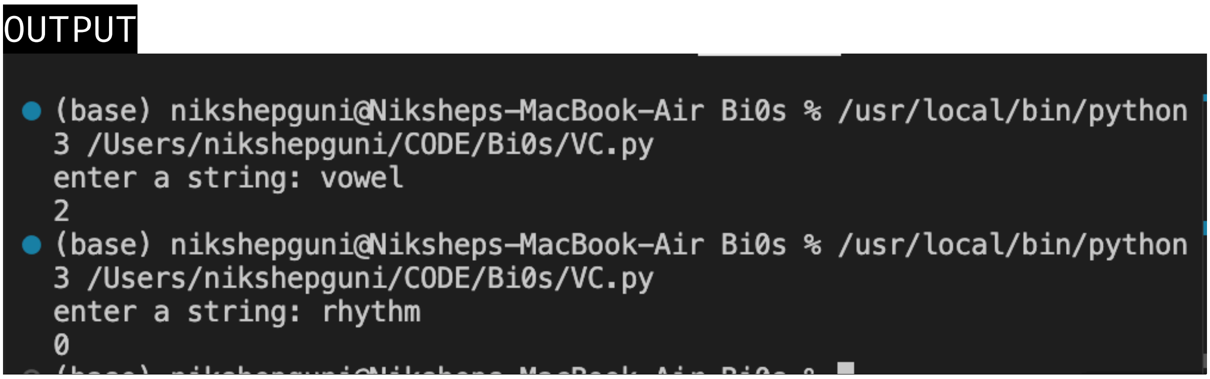
CODE



```python
def count_vowels(s):
    count = 0
    for char in s:
        if char in 'aeiou':
            count += 1
    return count

s = str(input("enter a string: "))
result = count_vowels(s)
print(result)
```

OUTPUT



```
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/VC.py
  enter a string: vowel
  2
● (base) nikshepguni@Niksheps-MacBook-Air Bi0s % /usr/local/bin/python
  3 /Users/nikshepguni/CODE/Bi0s/VC.py
  enter a string: rhythm
  0
  (base) nikshepguni@Niksheps-MacBook-Air Bi0s %
```