ASSIGNMENT-4

NAME: T. Nikhil Kumar Reddy

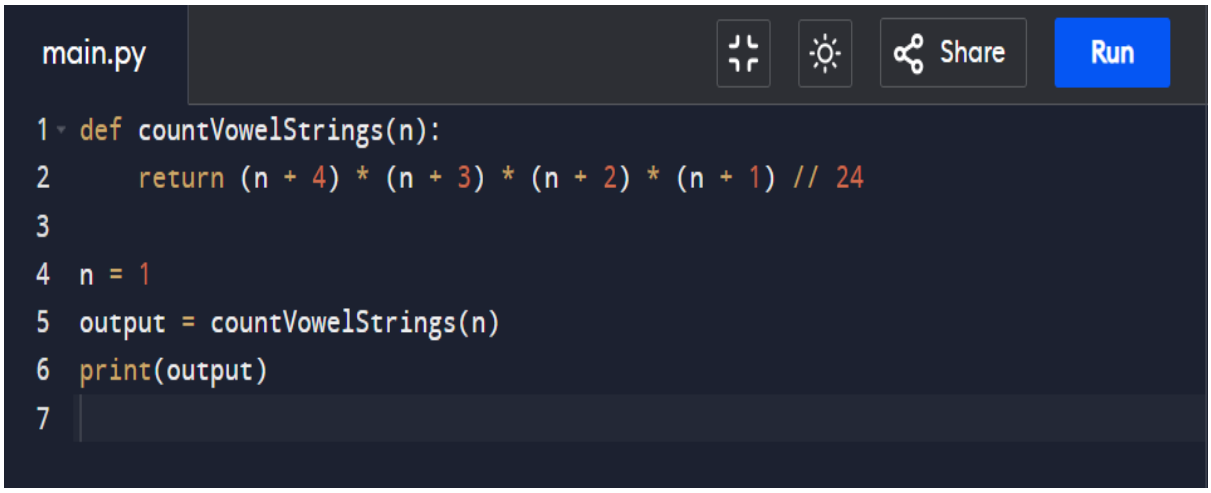
REG-NO: 192372024

SUBJECT: Python

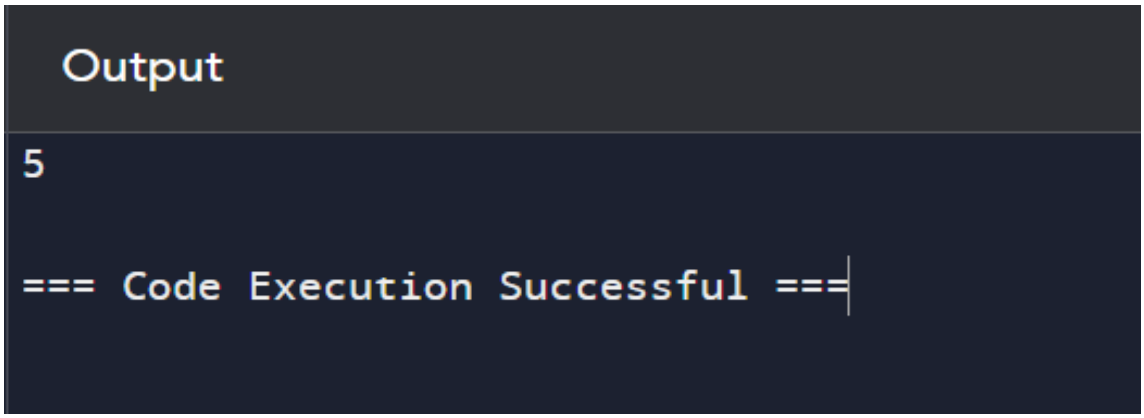
CODE: CSA0898

1.Given an integer n, return the number of strings of length n that consist only of vowels (a, e, i, o, u) and are lexicographically sorted. A string s is lexicographically sorted if for all valid i, s[i] is the same as or comes before s[i+1] in the alphabet.

INPUT:



OUTPUT:



2.Given two binary strings a and b, return their sum as a binary string. • a and b consist only of '0' or '1' characters. • Each string does not contain leading zeros except for the zero itself.

INPUT:

main.py

Share Run

```
1  def addBinary(a, b):
2      return bin(int(a, 2) + int(b, 2))[2:]
3
4  a = "11"
5  b = "1"
6  print(addBinary(a, b))
7
```

OUTPUT:

Output

```
100

=== Code Execution Successful ===
```

3. . Basic Calculator II Given a string s which represents an expression, evaluate this expression and return its value. The integer division should truncate toward zero. You may assume that the given expression is always valid. All intermediate results will be in the range of [-231, 231 - 1]. • s consists of integers and operators ('+', '-', '*', '/') separated by some number of spaces. • s represents a valid expression. • All the integers in the expression are non-negative integers in the range [0, 231 - 1]. The answer is guaranteed to fit in a 32-bit integer. Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as eval().

INPUT:

main.py

Share Run

```
class Solution:
    def calculate(self, s: str) -> int:
        stack = []
        num = 0
        sign = '+'
        for i, char in enumerate(s):
            if char.isdigit():
                num = num * 10 + int(char)
            if (not char.isdigit() and char != ' ') or i == len(s) - 1:
                if sign == '+':
                    stack.append(num)
                elif sign == '-':
                    stack.append(-num)
                elif sign == '*':
                    stack[-1] = stack[-1] * num
                elif sign == '/':
                    stack[-1] = int(stack[-1] / num)
                sign = char
                num = 0
        return sum(stack)
s = "3+2*2"
solution = Solution()
output = solution.calculate(s)
print(output)
```

OUTPUT:

Output

```
7

=== Code Execution Successful ===
```

4. Raju, has again started troubling people in your city. The people have turned on to you for getting rid of Raju. Raju presents to you a number consisting of numbers from 0 to 9 characters. He wants you to reverse it from the final answer such that the number becomes Mirror number. A Mirror is a number which equals its reverse. The hope of people are on you so you have to solve the riddle. You have to tell if some number exists which you would reverse to convert the number into Mirror Sample

INPUT:

main.py

Share Run

```
1 def reverse_number(num):
2     return int(str(num)[::-1])
3
4 num = int(input("Enter the number: "))
5 mirror_image = reverse_number(num)
6 print("Mirror image:", mirror_image)
7
```

OUTPUT:

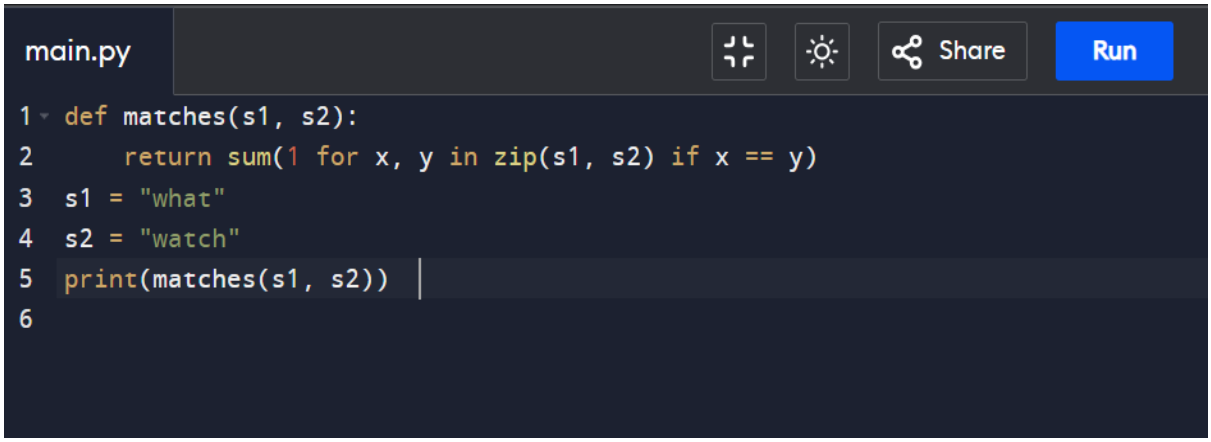
Output

```
Enter the number: 123456
Mirror image: 654321

=== Code Execution Successful ===
```

5. Write a python function called matches that takes two strings as arguments and returns how many matches there are between the strings. A match is where the two strings have the same character at the same index.

INPUT:



OUTPUT: