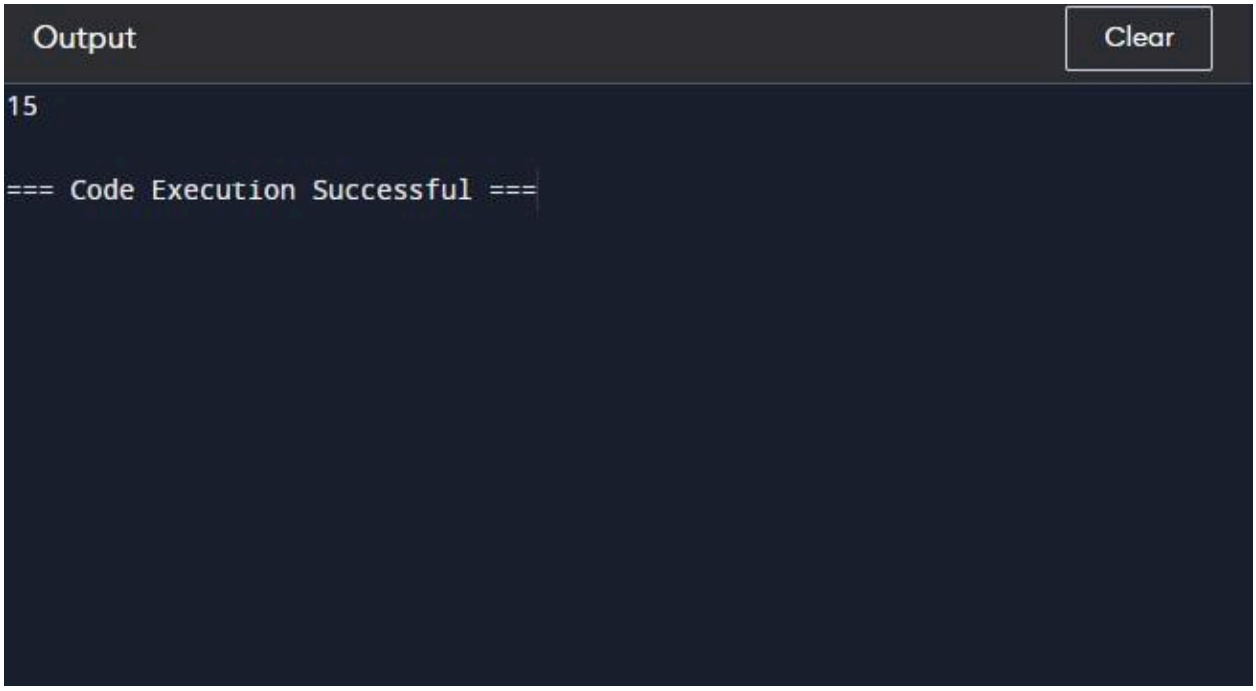
ASSIGNMENT-4

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.

Test Cases: 1.Input: n = 1 Output: 5
Explanation: The 5 sorted strings that consist of vowels only are ["a","e","i","o","u"]. 2.Input: n = 2 Output: 15
Explanation: The 15 sorted strings that consist of vowels only are ["aa","ae","ai","ao","au","ee","ei","eo","eu","ii","io","iu","oo","ou","uu"]. Note that "ea" is not a valid string since 'e' comes after 'a' in the alphabet.
3. Input: n = 33 Output: 66045
4.n=-5 5.n=10

Python code for this question:

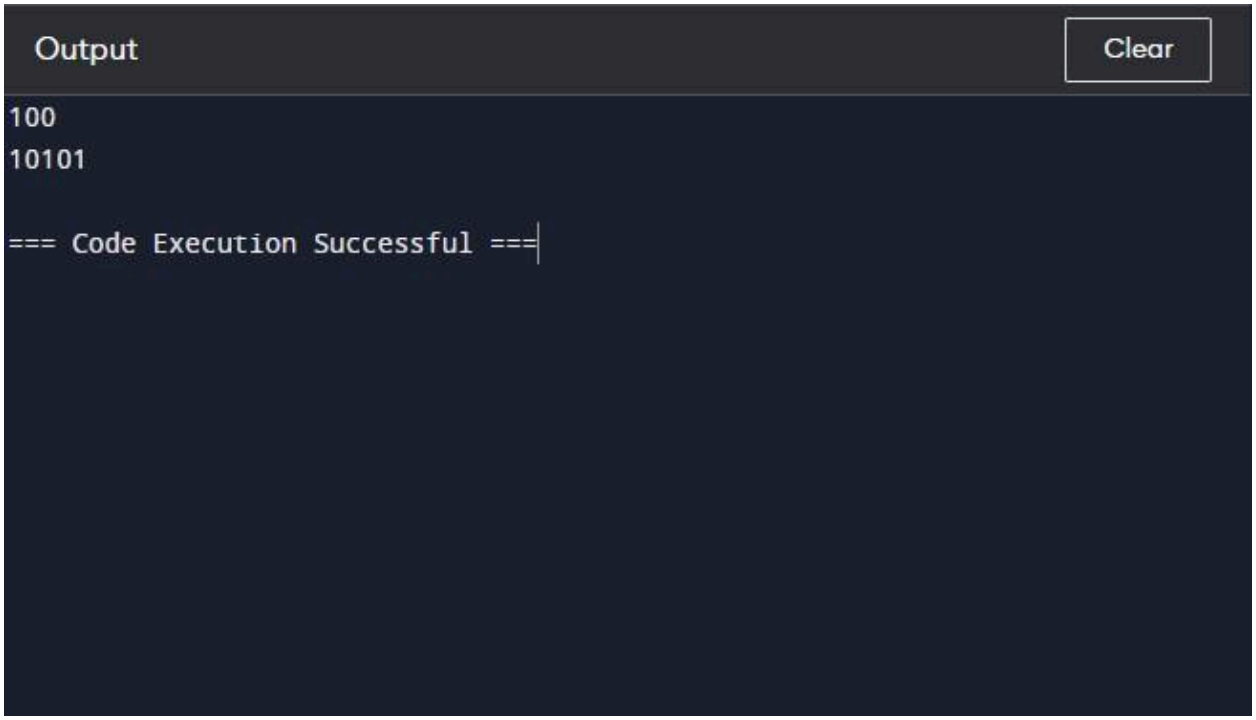
```
def countVowelStrings(n):
    dp = [[1] * 5 for _ in range(n)]

    for i in range(1, n):
        for j in range(5):
            dp[i][j] = sum(dp[i-1][j:])

    return sum(dp[-1])

print(countVowelStrings(2))  # Output: 15
```



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.
Test cases: 1.Input: a = "11", b = "1" Output: "100"
2.Input: a = "1010", b = "1011" Output: "10101"
3.a= “1111”, b= “1010”
4.a= “101101”, b= “1100”
5.a= “1011” b= “1111”

Python code for this question:

```
def addBinary(a, b):
   return bin(int(a, 2) + int(b, 2))[2:]
print(addBinary("11", "1"))
print(addBinary("1010", "1011"))
```

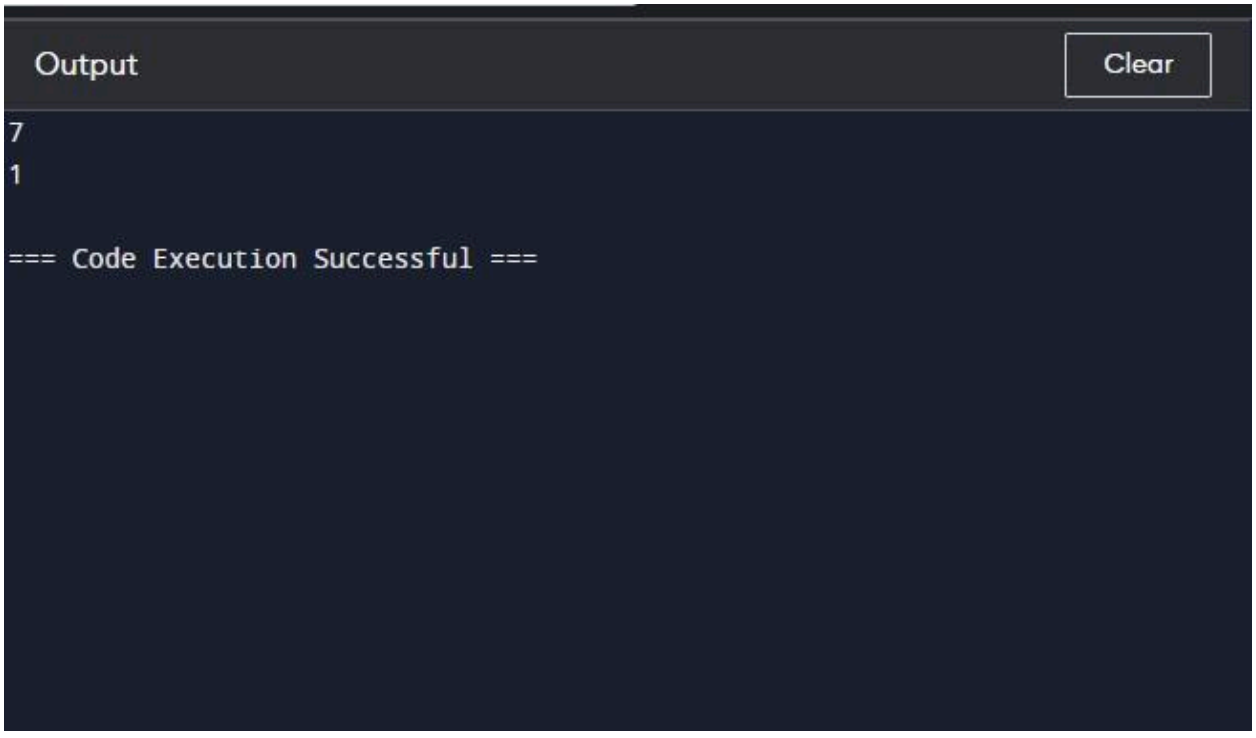


3. Basic Calculator II Given a string s which represents an expression, evaluate this expression and return its value. The integer division should truncate toward zero. You may assume that the given expression is always valid. All intermediate results will be in the range of [-231, 231 - 1]. • s consists 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().
Test cases: 1.Input: s = "3+2*2" Output: 7
2.Input: s = " 3/2 " Output: 1
3.Input: s = " 3+5 / 2 " Output: 5
4.s= “-1+5” 5.s= “2+3+5”

Python code for this question:

Class solution:
```
def calculate(s: str) -> int:
    s = s.replace(" ", "")
    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)

s1="3+2*2"
s2="3/2"
print(solution().calculate(s1))
print(solution().calculate(s2))
```

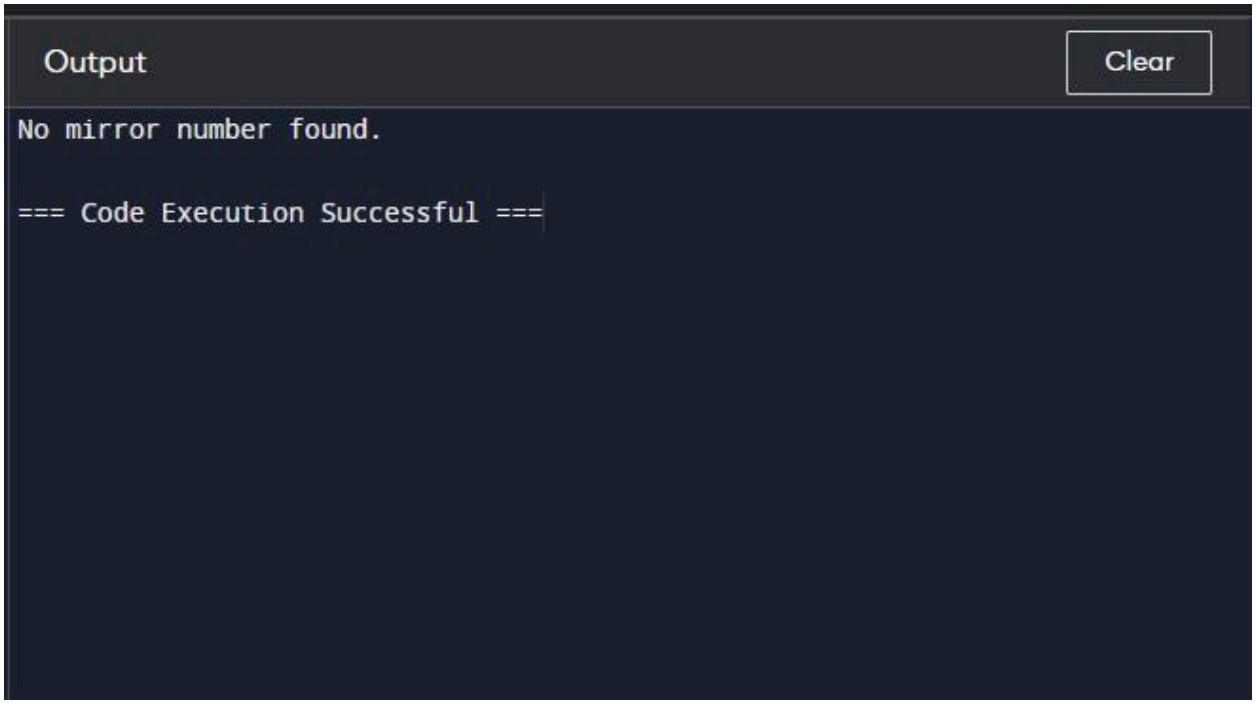


4. Raju, has again started troubling people in your city. The people have turned on to you for getting rid of Raju. Raju presents to you a number consisting of numbers from 0 to 9 characters. He wants you to reverse it from the final answer such that the number becomes Mirror number.

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: Enter the number: 123456
Sample output: Mirror image: 654321
Test cases: 1. Sell123
2. 5489236
3. Abc-abc
4. %$$$$^&
5. -123456

Python code for this question:

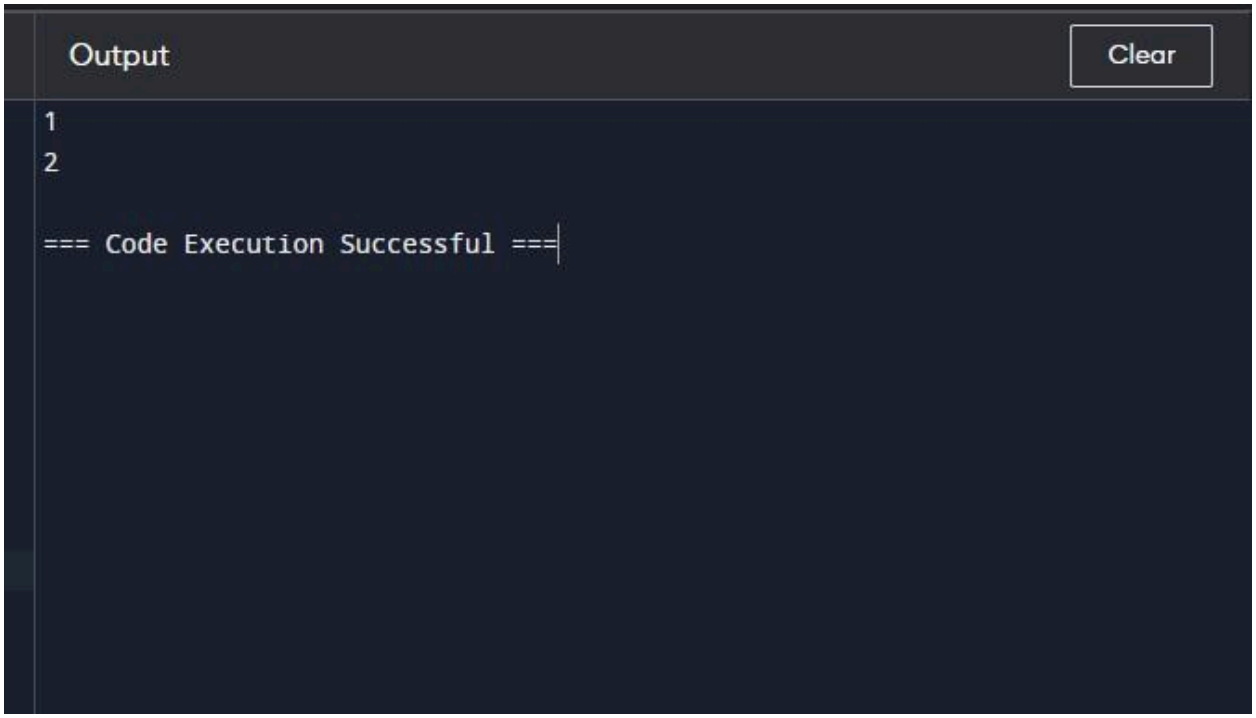
```
def find_mirror_number(number):
    reversed_number = int(str(number)[::-1])
    if number == reversed_number:
        return True
    else:
        return False

number = 123456
is_mirror = find_mirror_number(number)
if is_mirror:
    print(f"Mirror image: {number}")
else:
    print("No mirror number found.")
```



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.
Test Cases: 1. Input: s1= “what” s2= “watch” Output: 1
2.Input: s1= “ ran” s2= “van”
3. Input : s1 = “ rain” s2 = “ turn”
4.Input : s1 = “ python” s2 = “py”
5. Input: s1= “man” s2= “women

Python code for this question:

```
def matches(s1, s2):
    count = 0
    min_len = min(len(s1), len(s2))

    for i in range(min_len):
        if s1[i] == s2[i]:
            count += 1

    return count

print(matches("what", "watch"))
print(matches("python", "py"))
```


Output
Clear
1
2
=== Code Execution Successful ===