

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. $n = 10$

```
AS4-1.py - C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-1.py (3.12.4)
File Edit Format Run Options Window Help
def countVowelStrings(n):
    return (n + 4) * (n + 3) * (n + 2) * (n + 1) // 24
n = 10
print(countVowelStrings(n))

IDLE Shell 3.12.4
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-1.py
5
>>>
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-1.py =====
15
>>>
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-1.py =====
66045
>>>
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-1.py =====
1
>>>
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-1.py =====
1001
>>>
```

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"

The screenshot shows a Python IDE with two windows. The left window, titled 'AS4-2.py', contains the following code:

```
def addbinary(a,b):
    x,y=int(a,2),int(b,2)
    while y:
        x,y=x^y,(x&y)<<1
    return bin(x)[2:]
a="1011"
b="1111"
print(addbinary(a,b))
```

The right window, titled 'IDLE Shell 3.12.4', shows the output of the program after multiple restarts. The output is as follows:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-2.py
100
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-2.py =====
10101
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-2.py =====
11001
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-2.py =====
111001
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-2.py =====
11010
>>> |
```

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"

```
def calculate(s):
    stack = []
    num = 0
    sign = '+'
    for i in range(len(s)):
        if s[i].isdigit():
            num = num * 10 + int(s[i])
        if s[i] in "+-*/" 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)
            num = 0
            sign = s[i]
    return sum(stack)
s = "2+3+5"
print(calculate(s))
```

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-3.py
>>> 7
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-3.py =====
>>> 1
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-3.py =====
>>> 5
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-3.py =====
>>> 4
===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-3.py =====
>>> 10
>>> |
```

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

```
def reverse_number(num):  
    return int(num[::-1])  
def is_mirror_number(num):  
    return num == reverse_number(num)  
number = "123456"  
mirror_image = reverse_number(number)  
print(f"Mirror image: {mirror_image}")
```

```
Python 3.12.4 (tags/v3.12.4:8e9a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]  
on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py  
Mirror image: 654321  
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py =====  
Traceback (most recent call last):  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 6, in <module>  
    mirror_image = reverse_number(number)  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 2, in reverse_number  
    return int(num[::-1])  
ValueError: invalid literal for int() with base 10: ' 321lleS'  
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py =====  
Mirror image: 6329845  
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py =====  
Traceback (most recent call last):  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 6, in <module>  
    mirror_image = reverse_number(number)  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 2, in reverse_number  
    return int(num[::-1])  
ValueError: invalid literal for int() with base 10: 'cba-cbA'  
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py =====  
Traceback (most recent call last):  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 6, in <module>  
    mirror_image = reverse_number(number)  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 2, in reverse_number  
    return int(num[::-1])  
ValueError: invalid literal for int() with base 10: '&$$$%'  
>>> ===== RESTART: C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py =====  
Traceback (most recent call last):  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 6, in <module>  
    mirror_image = reverse_number(number)  
  File "C:/Users/siddu/AppData/Local/Programs/Python/Python312/AS4-4.py", line 2, in reverse_number  
    return int(num[::-1])  
ValueError: invalid literal for int() with base 10: ' 321lleS'
```

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:

- Input: s1= “ what” s2= “ watch” Output: 1
- Input: s1= “ ran” s2= “ van”
- Input: s1= “ rain” s2= “ turn”
- Input: s1= “ python” s2= “ py”
- Input: s1= “ man” s2= “ women

