**Python Advance Programming Assignment-05**

**1. Create a function that takes a number n (integer greater than zero) as an argument, and returns 2 if n is odd and 8 if n is even.**

You can only use the following arithmetic operators: addition of numbers +, subtraction of numbers -, multiplication of number \*, division of number /, and exponentiation \*\*.

You are not allowed to use any other methods in this challenge (i.e. no if statements, comparison operators, etc).

Examples

f(1) ➞ 2

f(2) ➞ 8

f(3) ➞ 2

In [14]:

**def** f(num):

arr **=** [8, 2]

num **=** int(num)

**return** arr[num **%** 2]

In [15]:

f(1)

Out[15]:

2

In [16]:

f(2)

Out[16]:

8

In [17]:

f(3)

Out[17]:

2

**2. Create a function that returns the majority vote in a list. A majority vote is an element that occurs > N/2 times in a list (where N is the length of the list).**

Examples

majority\_vote(["A", "A", "B"]) ➞ "A"

majority\_vote(["A", "A", "A", "B", "C", "A"]) ➞ "A"

majority\_vote(["A", "B", "B", "A", "C", "C"]) ➞ None

In [58]:

**def** majority\_vote(l):

d**=**{}

**for** i **in** l:

d[i]**=**l**.**count(i)

maximum**=**max(d**.**items(),key**=lambda** a:a[1])[1]

**if** maximum**>**len(l)**/**2:

print(maximum)

**else**:

print(**None**)

In [59]:

majority\_vote(["A", "A", "B"])

2

In [60]:

majority\_vote(["A", "A", "A", "B", "C", "A"])

4

In [61]:

majority\_vote(["A", "B", "B", "A", "C", "C"])

None

**3. Create a function that takes a string txt and censors any word from a given list lst. The text removed must be replaced by the given character char.**

Examples

censor\_string("Today is a Wednesday!", ["Today", "a"], "-") ➞ "----- is - Wednesday!"

censor\_string("The cow jumped over the moon.", ["cow", "over"], "*"), "The \*\** jumped *\*\** the moon.")

censor\_string("Why did the chicken cross the road?", ["Did", "chicken", "road"], "*") ➞ "Why \*\** the **\*\*\*** cross the *\*\**?"

In [90]:

**def** censor\_string(string1,lst,char):

a1**=**string1**.**split()

index**=**0

**for** i **in** a1:

**for** j **in** lst:

**if** i **==** j:

a1[index]**=**char**\***len(i)

index**+=**1

**return** ' '**.**join(a1)

In [91]:

censor\_string("Today is a Wednesday!", ["Today", "a"], "-")

Out[91]:

'----- is - Wednesday!'

In [92]:

censor\_string("The cow jumped over the moon.", ["cow", "over"], "\*")

Out[92]:

'The \*\*\* jumped \*\*\*\* the moon.'

In [93]:

censor\_string("Why did the chicken cross the road?", ["Did", "chicken", "road"], "\*")

Out[93]:

'Why did the \*\*\*\*\*\*\* cross the road?'

**4. In mathematics a Polydivisible Number (or magic number) is a number in a given number base with digits abcde... that has the following properties:**

* Its first digit a is not 0.
* The number formed by its first two digits ab is a multiple of 2.
* The number formed by its first three digits abc is a multiple of 3.
* The number formed by its first four digits abcd is a multiple of 4.

Create a function which takes an integer n and returns True if the given number is a Polydivisible Number and False otherwise.

Examples

is\_polydivisible(1232) ➞ True

1 / 1 = 1

12 / 2 = 6

123 / 3 = 41

1232 / 4 = 308

is\_polydivisible(123220 ) ➞ False

1 / 1 = 1

12 / 2 = 6

123 / 3 = 41

1232 / 4 = 308

12322 / 5 = 2464.4 # Not a Whole Number

123220 /6 = 220536.333... # Not a Whole Number

**5. Create a function that takes a list of numbers and returns the sum of all prime numbers in the list.**

Examples

sum\_primes([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]) ➞ 17

sum\_primes([2, 3, 4, 11, 20, 50, 71]) ➞ 87

sum\_primes([]) ➞ None

In [116]:

**def** sum\_primes(lon):

sum**=**0

**if** len(lon)**>**0:

**for** num **in** lon:

i**=**1

**for** i **in** range(2,num):

**if**(int(num**%i**==0)):

i**=**num

**break**;

**if** i **is** **not** num:

sum**+=**num

print(sum)

**else**:

print(**None**)

In [117]:

sum\_primes([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

17

In [118]:

sum\_primes([2, 3, 4, 11, 20, 50, 71])

87

In [119]:

sum\_primes([])

None