**ADVANCED PROGRAMMING ASSIGNMENT\_5 -SUBMITTED BY SAMUEL DEVDAS**

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

Ans.

def f(inp):

start=2

end=8

for i in range(inp):

start,end=end,start

return end

f(3)

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

Ans.

def majority\_vote(inp):

count={}

for i in inp:

count[i]=inp.count(i)

for j in count:

if max(count.values())==min(count.values()):

print(None)

break

elif count[j]==max(count.values()):

return(j)

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

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 \*\*\*\*?"

Ans.

def censor\_string(inp,cen,replace):

split\_inp=inp.split()

censored=[]

for elem in split\_inp:

if elem in cen:

for i in elem:

elem=elem.replace(i,replace)

censored.append(elem)

else:

censored.append(elem)

output=''

for j in censored:

output+=' '+j

return(output.lstrip())

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

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

Ans.

def is\_polydivisible(inp):

str\_inp=str(inp)

separate=[]

for i in range(len(str\_inp)+1):

separate.append(str\_inp[0:i])

for j in range(1,len(separate)):

if int(separate[j])%j==0:

if separate[-1]!=separate[j]:

continue

elif separate[-1]==separate[j]:

if int(separate[j])%j==0:

print(True)

else:

print(False)

break

is\_polydivisible(1232)

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

Ans.

import sympy

def sum\_primes(inp):

primes=[]

if len(inp)==0:

print(None)

else:

for i in inp:

if sympy.isprime(i):

primes.append(i)

return sum(primes)

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