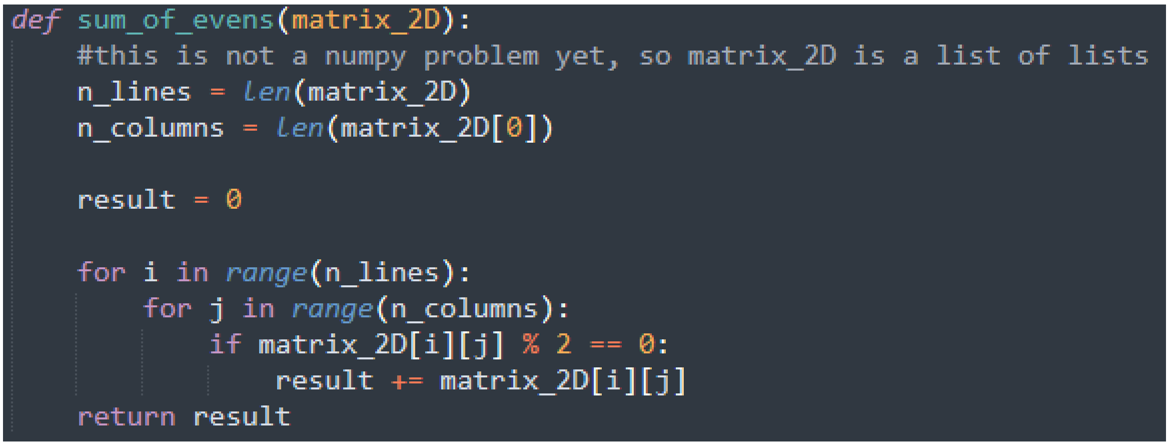
Problem 11

Create a function that returns the sum of all even elements in a 2D matrix.



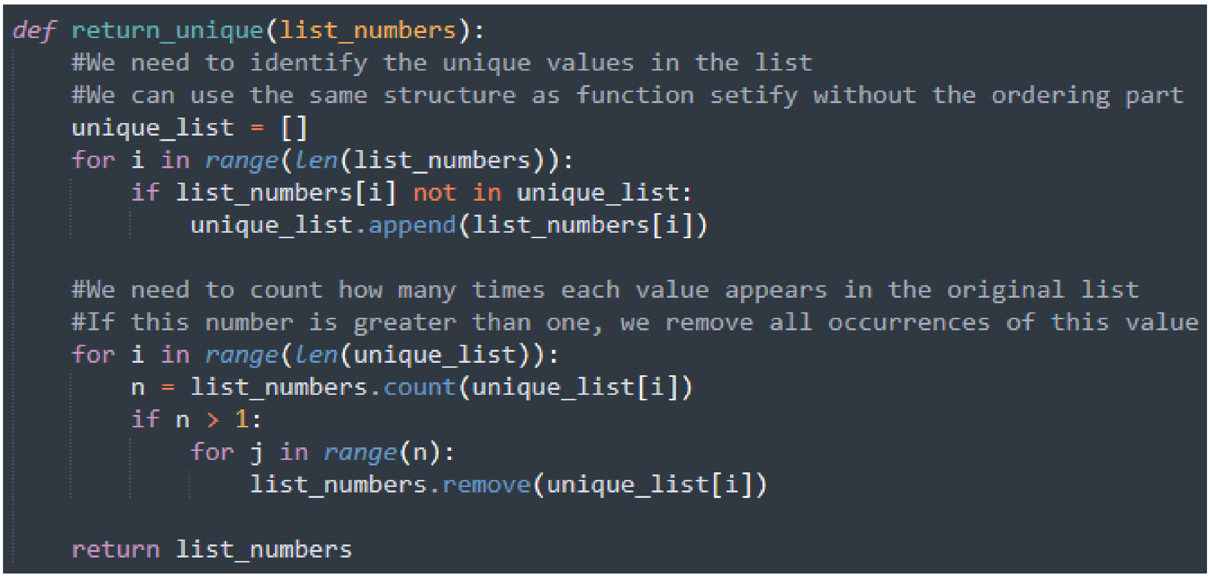
```python
def sum_of_evens(matrix_2D):
    #this is not a numpy problem yet, so matrix_2D is a list of lists
    n_lines = len(matrix_2D)
    n_columns = len(matrix_2D[0])

    result = 0

    for i in range(n_lines):
        for j in range(n_columns):
            if matrix_2D[i][j] % 2 == 0:
                result += matrix_2D[i][j]
    return result
```

Problem 12

In each input list, every number repeats at least once, except for two. Write a function that returns the two unique numbers.



```python
def return_unique(list_numbers):
    #We need to identify the unique values in the list
    #We can use the same structure as function setify without the ordering part
    unique_list = []
    for i in range(len(list_numbers)):
        if list_numbers[i] not in unique_list:
            unique_list.append(list_numbers[i])

    #We need to count how many times each value appears in the original list
    #If this number is greater than one, we remove all occurrences of this value
    for i in range(len(unique_list)):
        n = list_numbers.count(unique_list[i])
        if n > 1:
            for j in range(n):
                list_numbers.remove(unique_list[i])

    return list_numbers
```

Problem 13

This is a reverse coding challenge. Normally you're given explicit directions with how to create a function. Here, you must generate your own function to satisfy the relationship between the inputs and outputs.

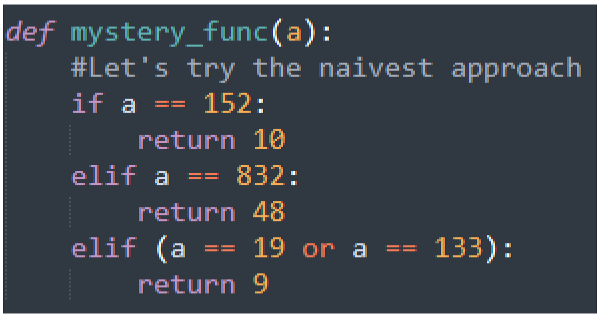
Your task is to create a function that, when fed the inputs below, produces the sample outputs shown.

mystery_func(152) ➞ 10

mystery_func(832) ➞ 48

mystery_func(19) ➞ 9

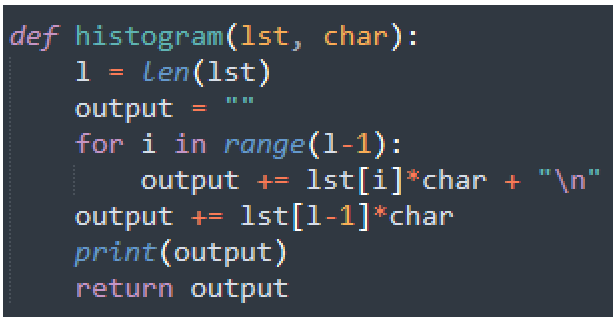
mystery_func(133) ➞ 9



```
def mystery_func(a):
    #Let's try the naivest approach
    if a == 152:
        return 10
    elif a == 832:
        return 48
    elif (a == 19 or a == 133):
        return 9
```

Problem 14

Build a function that creates histograms. Every bar needs to be on a new line and its length corresponds to the numbers in the list passed as an argument. The second argument of the function represents the character that needs to be used.



```
def histogram(lst, char):
    l = len(lst)
    output = ""
    for i in range(l-1):
        output += lst[i]*char + "\n"
    output += lst[l-1]*char
    print(output)
    return output
```

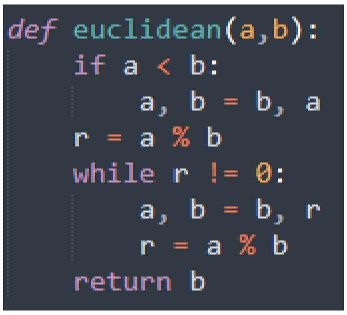
Problem 15

This challenge will deal further with writing recursive functions by covering the Euclidean Algorithm. The "Euclidean Algorithm" is a method for finding the greatest common divisor (GCD) of two numbers. It was originally described by the Greek mathematician Euclid.
Algorithm

For the sake of simplicity I'll refer to the first number as "a", the second number as "b", and the remainder as "r". The algorithm can be broken down into four steps:

```
Ensure that "a" >= "b". If "a" < "b", swap them.
Find the remainder. Divide "a" by "b" and set "r" as the remainder.
Is "r" zero? If so terminate the function and return "b" (the second
number).
Set "a" = "b" and "b" = "r" and start the algorithm over again.
```

Create a recursive function that returns the GCD between two positive numbers using the Euclidean Algorithm.



```python
def euclidean(a,b):
    if a < b:
        a, b = b, a
    r = a % b
    while r != 0:
        a, b = b, r
        r = a % b
    return b
```