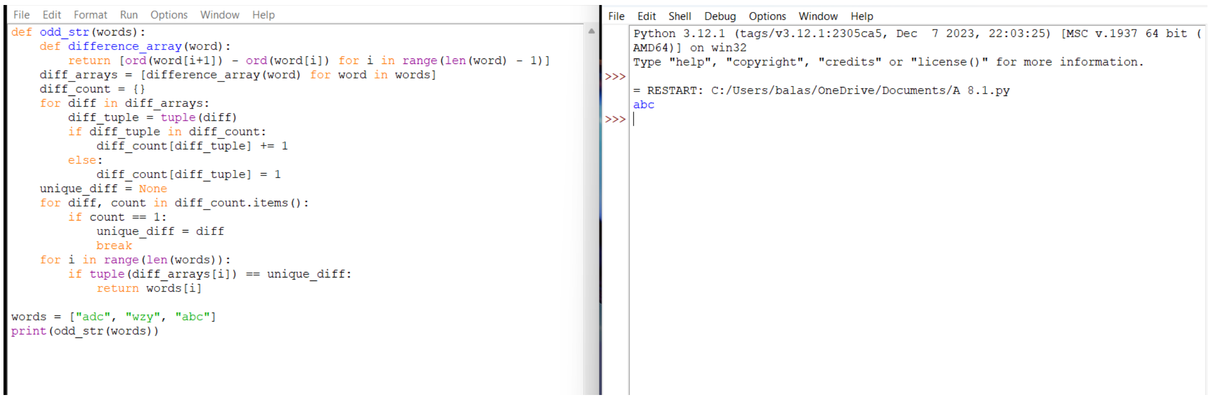
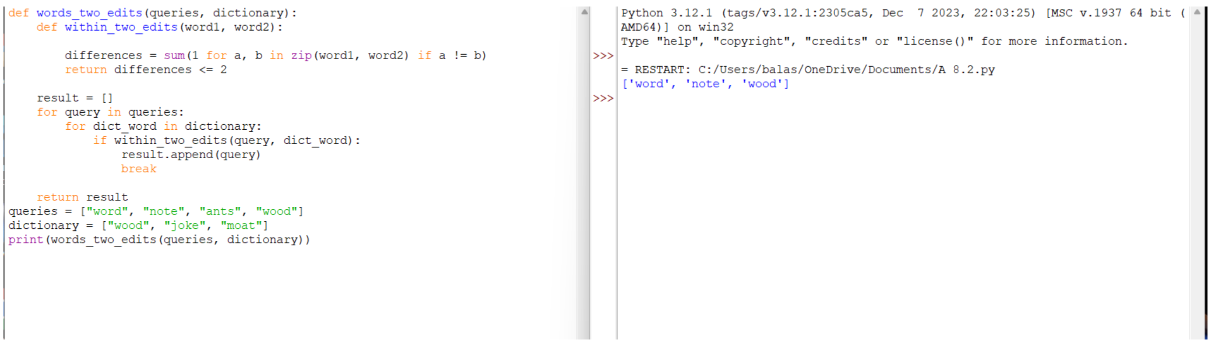
1. Odd String Difference You are given an array of equal-length strings words. Assume that the length of each string is n.Each string words[i] can be converted into a difference integer array difference[i] of length n - 1 where difference[i][j] = words[i][j+1] - words[i][j] where 0 <= j <= n - 2. Note that the difference between two letters is the difference between their positions in the alphabet i.e. the position of 'a' is 0, 'b' is 1, and 'z' is 25.


File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

```python
def odd_str(words):
    def difference_array(word):
        return [ord(word[i+1]) - ord(word[i]) for i in range(len(word) - 1)]
    diff_arrays = [difference_array(word) for word in words]
    diff_count = {}
    for diff in diff_arrays:
        diff_tuple = tuple(diff)
        if diff_tuple in diff_count:
            diff_count[diff_tuple] += 1
        else:
            diff_count[diff_tuple] = 1
    unique_diff = None
    for diff, count in diff_count.items():
        if count == 1:
            unique_diff = diff
            break
    for i in range(len(words)):
        if tuple(diff_arrays[i]) == unique_diff:
            return words[i]

words = ["adc", "wzy", "abc"]
print(odd_str(words))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 8.1.py
abc
>>>
```

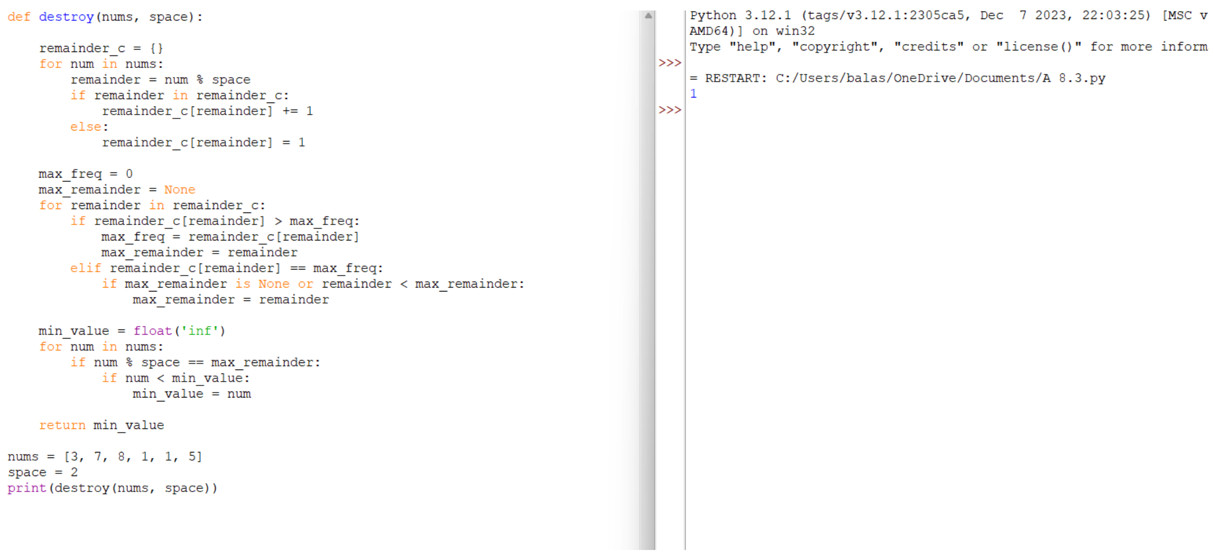
2. Words Within Two Edits of Dictionary You are given two string arrays, queries and dictionary. All words in each array comprise of lowercase English letters and have the same length.In one edit you can take a word from queries, and change any letter in it to any other letter. Find all words from queries that, after a maximum of two edits, equal some word from dictionary. Return a list of all words from queries, that match with some word from dictionary after a maximum of two edits. Return the words in the same order they appear in queries.



```python
def words_two_edits(queries, dictionary):
    def within_two_edits(word1, word2):

        differences = sum(1 for a, b in zip(word1, word2) if a != b)
        return differences <= 2

    result = []
    for query in queries:
        for dict_word in dictionary:
            if within_two_edits(query, dict_word):
                result.append(query)
                break

    return result
queries = ["word", "note", "ants", "wood"]
dictionary = ["wood", "joke", "moat"]
print(words_two_edits(queries, dictionary))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 8.2.py
['word', 'note', 'wood']
>>>
```

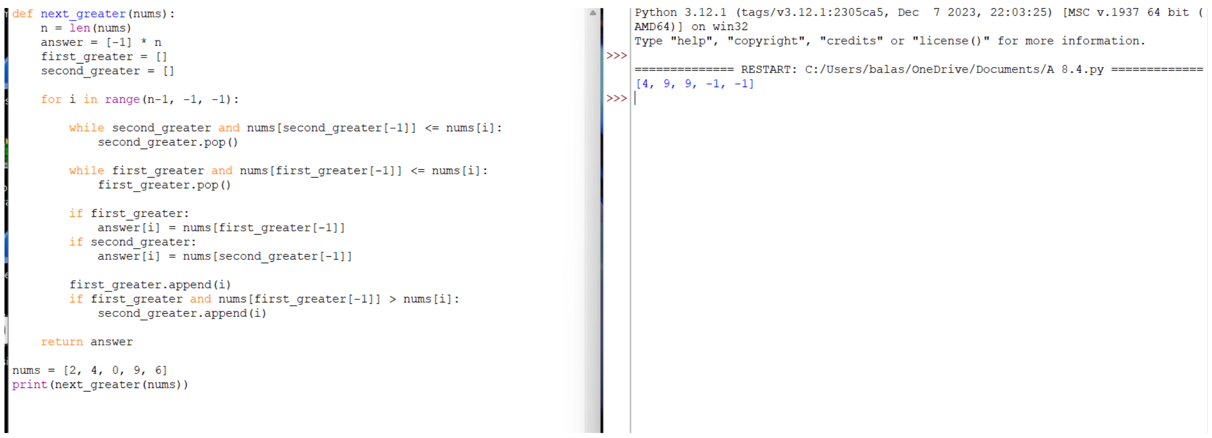
3. . Destroy Sequential Targets You are given a 0-indexed array nums consisting of positive integers, representing targets on a number line. You are also given an integer space. You have a machine which can destroy targets. Seeding the machine with some nums[i] allows it to destroy all targets with values that can be represented as nums[i] + c * space, where c is any non-negative integer. You want to destroy the maximum number of targets in nums. Return the minimum value of nums[i] you can seed the machine with to destroy the maximum number of targets.



```
def destroy(nums, space):

    remainder_c = {}
    for num in nums:
        remainder = num % space
        if remainder in remainder_c:
            remainder_c[remainder] += 1
        else:
            remainder_c[remainder] = 1

    max_freq = 0
    max_remainder = None
    for remainder in remainder_c:
        if remainder_c[remainder] > max_freq:
            max_freq = remainder_c[remainder]
            max_remainder = remainder
        elif remainder_c[remainder] == max_freq:
            if max_remainder is None or remainder < max_remainder:
                max_remainder = remainder

    min_value = float('inf')
    for num in nums:
        if num % space == max_remainder:
            if num < min_value:
                min_value = num

    return min_value

nums = [3, 7, 8, 1, 1, 5]
space = 2
print(destroy(nums, space))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more inform
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 8.3.py
1
>>>
```

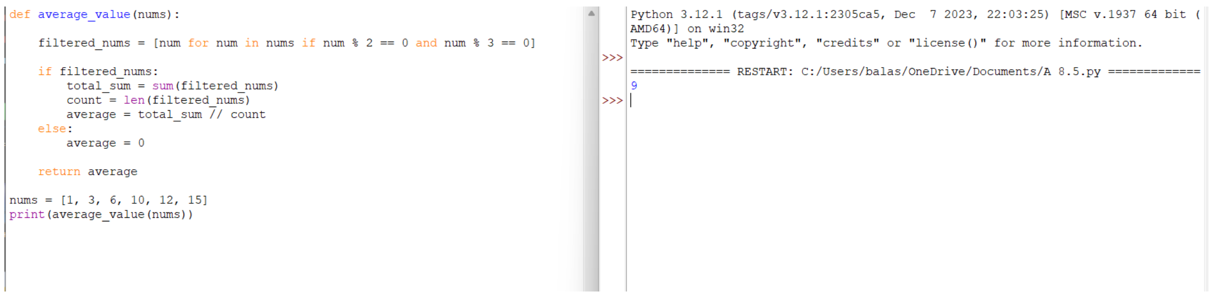
4. Next Greater Element IV You are given a 0-indexed array of non-negative integers nums. For each integer in nums, you must find its respective second largest INTEGER.



```
def next_greater(nums):
    n = len(nums)
    answer = [-1] * n
    first_greater = []
    second_greater = []

    for i in range(n-1, -1, -1):

        while second_greater and nums[second_greater[-1]] <= nums[i]:
            second_greater.pop()

        while first_greater and nums[first_greater[-1]] <= nums[i]:
            first_greater.pop()

        if first_greater:
            answer[i] = nums[first_greater[-1]]
        if second_greater:
            answer[i] = nums[second_greater[-1]]

        first_greater.append(i)
        if first_greater and nums[first_greater[-1]] > nums[i]:
            second_greater.append(i)

    return answer

nums = [2, 4, 0, 9, 6]
print(next_greater(nums))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:/Users/balas/OneDrive/Documents/A 8.4.py =============
[4, 9, 9, -1, -1]
>>>
```

5. Average Value of Even Numbers That Are Divisible by Three Given an integer array nums of positive integers, return the average value of all even integers that are divisible by 3. Note that the average of n elements is the sum of the n elements divided by n and rounded down to the nearest integer.

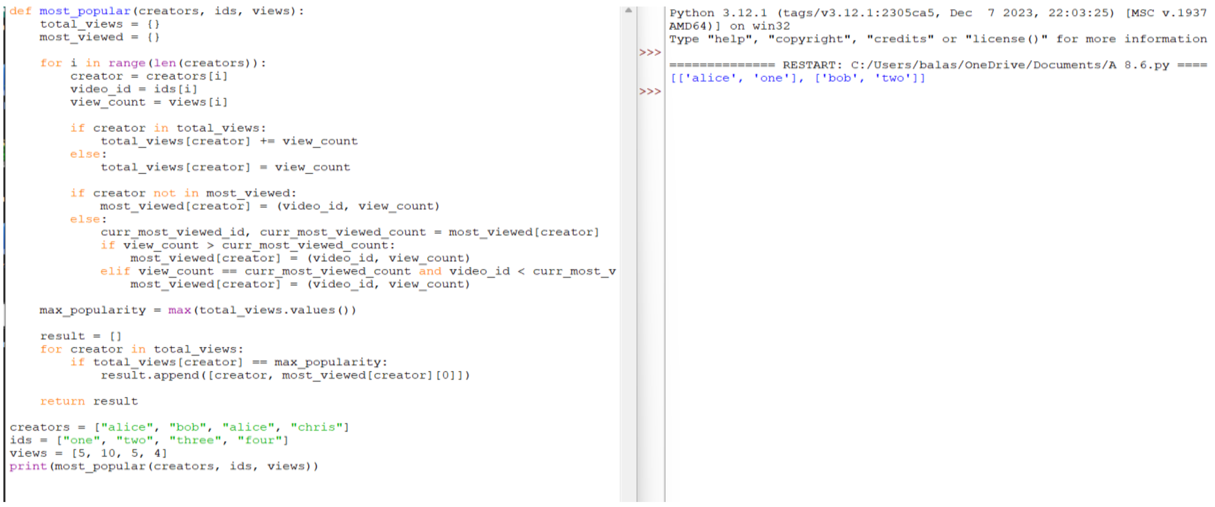


```
def average_value(nums):

    filtered_nums = [num for num in nums if num % 2 == 0 and num % 3 == 0]

    if filtered_nums:
        total_sum = sum(filtered_nums)
        count = len(filtered_nums)
        average = total_sum // count
    else:
        average = 0

    return average

nums = [1, 3, 6, 10, 12, 15]
print(average_value(nums))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:/Users/balas/OneDrive/Documents/A 8.5.py =============
9
>>>
```

6. Most Popular Video Creator You are given two string arrays creators and ids, and an integer array views, all of length n. The ith video on a platform was created by creator[i], has an id of ids[i], and has views[i] views. The popularity of a creator is the sum of the number of views on all of the creator's videos. Find the creator with the highest popularity and the id of their most viewed video. ● If multiple creators have the highest popularity, find all of them. ● If multiple videos have the highest view count for a creator, find the lexicographically smallest id. Return a 2D array of strings

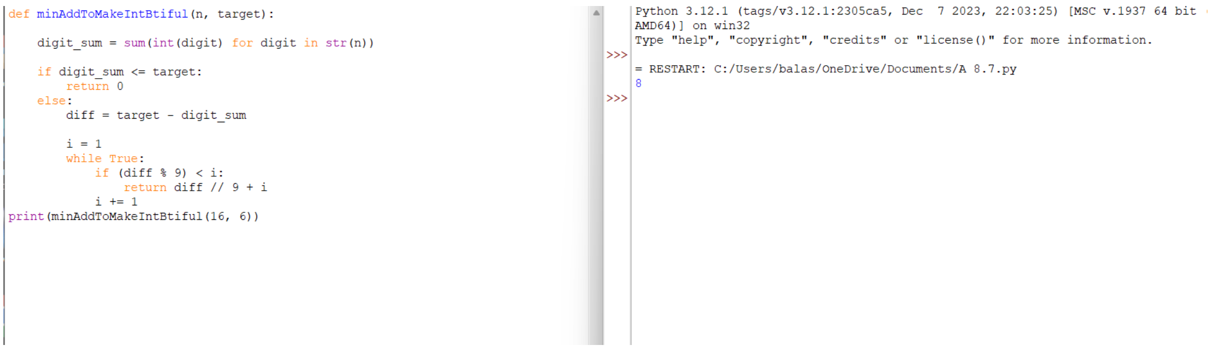
answer where answer[i] = [creatori, idi] means that creatori has the highest popularity and idi is the id of their most popular video. The answer can be returned in any order.



```python
def most_popular(creators, ids, views):
    total_views = {}
    most_viewed = {}

    for i in range(len(creators)):
        creator = creators[i]
        video_id = ids[i]
        view_count = views[i]

        if creator in total_views:
            total_views[creator] += view_count
        else:
            total_views[creator] = view_count

        if creator not in most_viewed:
            most_viewed[creator] = (video_id, view_count)
        else:
            curr_most_viewed_id, curr_most_viewed_count = most_viewed[creator]
            if view_count > curr_most_viewed_count:
                most_viewed[creator] = (video_id, view_count)
            elif view_count == curr_most_viewed_count and video_id < curr_most_v
                most_viewed[creator] = (video_id, view_count)

    max_popularity = max(total_views.values())

    result = []
    for creator in total_views:
        if total_views[creator] == max_popularity:
            result.append([creator, most_viewed[creator][0]])

    return result

creators = ["alice", "bob", "alice", "chris"]
ids = ["one", "two", "three", "four"]
views = [5, 10, 5, 4]
print(most_popular(creators, ids, views))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
============== RESTART: C:/Users/balas/OneDrive/Documents/A 8.6.py ====
[['alice', 'one'], ['bob', 'two']]
>>>
```

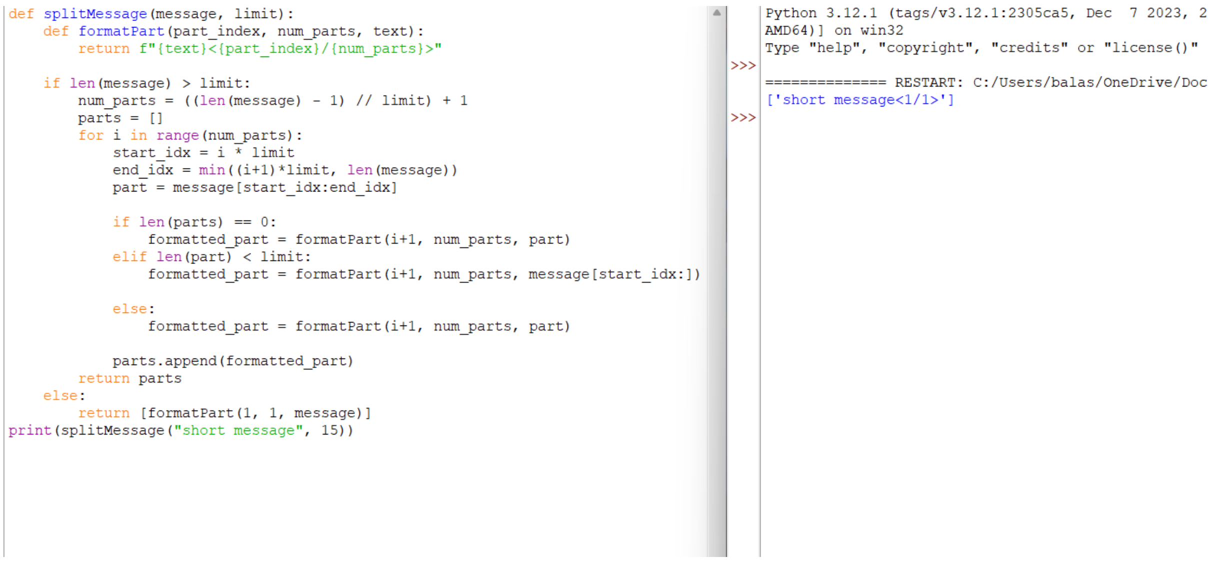
7. Minimum Addition to Make Integer Beautiful You are given two positive integers n and target. An integer is considered beautiful if the sum of its digits is less than or equal to target. Return the minimum non-negative integer x such that n + x is beautiful. The input will be generated such that it is always possible to make n beautiful.



```python
def minAddToMakeIntBtiful(n, target):

    digit_sum = sum(int(digit) for digit in str(n))

    if digit_sum <= target:
        return 0
    else:
        diff = target - digit_sum

        i = 1
        while True:
            if (diff % 9) < i:
                return diff // 9 + i
            i += 1
print(minAddToMakeIntBtiful(16, 6))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 8.7.py
8
>>>
```

8. . Split Message Based on Limit You are given a string, message, and a positive integer, limit. You must split message into one or more parts based on limit. Each resulting part should have the suffix "", where "b" is to be replaced with the total number of parts and "a" is to be replaced with the index of the part, starting from 1 and going up to b. Additionally, the length of each resulting part (including its suffix) should be equal to limit, except for the last part whose length can be at most limit. The resulting parts should be formed such that when their suffixes are removed and they are all concatenated in order, they should be equal to message. Also, the result should contain as few parts as possible. Return the parts message would be split into as an array of strings. If it is impossible to split message as required, return an empty array.



```
def splitMessage(message, limit):
    def formatPart(part_index, num_parts, text):
        return f"{text}<{part_index}/{num_parts}>"

    if len(message) > limit:
        num_parts = ((len(message) - 1) // limit) + 1
        parts = []
        for i in range(num_parts):
            start_idx = i * limit
            end_idx = min((i+1)*limit, len(message))
            part = message[start_idx:end_idx]

            if len(parts) == 0:
                formatted_part = formatPart(i+1, num_parts, part)
            elif len(part) < limit:
                formatted_part = formatPart(i+1, num_parts, message[start_idx:])

            else:
                formatted_part = formatPart(i+1, num_parts, part)

            parts.append(formatted_part)
        return parts
    else:
        return [formatPart(1, 1, message)]
print(splitMessage("short message", 15))
```

```
    Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 2
    AMD64)] on win32
    Type "help", "copyright", "credits" or "license()"
>>>
    ============== RESTART: C:/Users/balas/OneDrive/Doc
    ['short message<1/1>']
>>>
```