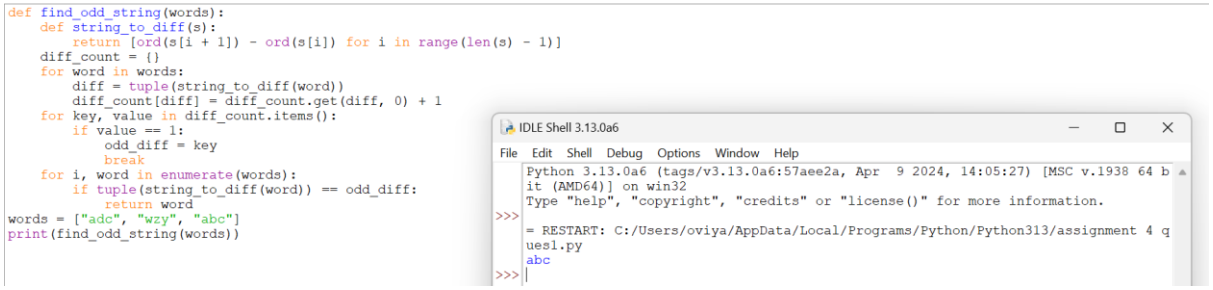
ASSIGNMENT-4

1. Odd String Difference

```
def find_odd_string(words):
    def string_to_diff(s):
        return [ord(s[i + 1]) - ord(s[i]) for i in range(len(s) - 1)]
    diff_count = {}
    for word in words:
        diff = tuple(string_to_diff(word))
        diff_count[diff] = diff_count.get(diff, 0) + 1
    for key, value in diff_count.items():
        if value == 1:
            odd_diff = key
            break
    for i, word in enumerate(words):
        if tuple(string_to_diff(word)) == odd_diff:
            return word
words = ["adc", "wzy", "abc"]
print(find_odd_string(words))
```

//**Time complexity= O(n^2)**

OUTPUT:



2. Words Within Two Edits of Dictionary

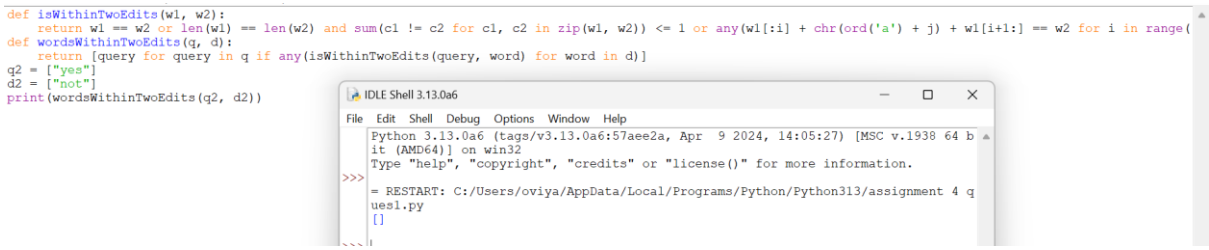
```
def isWithinTwoEdits(w1, w2):
    return w1 == w2 or len(w1) == len(w2) and sum(c1 != c2 for c1, c2 in zip(w1, w2)) <= 1 or any(w1[:i] + chr(ord('a') + j) + w1[i+1:] == w2 for i in range(len(w1)) for j in range(26))
def wordsWithinTwoEdits(q, d):
    return [query for query in q if any(isWithinTwoEdits(query, word) for word in d)]
q2 = ["yes"]
```

d2 = ["not"]

print(wordsWithinTwoEdits(q2, d2))

//**Time complexity= O(n)**

OUTPUT:

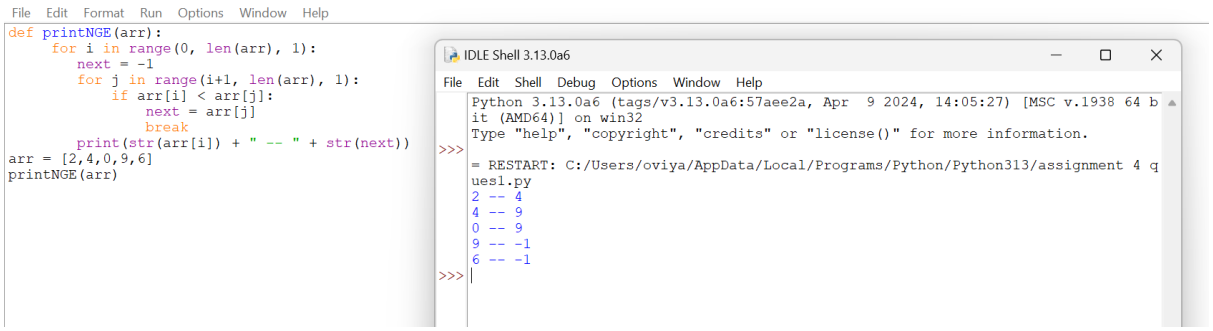


3. Next Greater Element IV

```
def printNGE(arr):
    for i in range(0, len(arr), 1):
        next = -1
        for j in range(i+1, len(arr), 1):
            if arr[i] < arr[j]:
                next = arr[j]
                break
        print(str(arr[i]) + " -- " + str(next))
arr = [2,4,0,9,6]
printNGE(arr)
```

//**Time complexity= O(n^2)**

OUTPUT:



4. Minimum Addition to Make Integer Beautiful

```
def digit_sum(num):
    return sum(int(digit) for digit in str(num))
def min_addition_to_make_beautiful(n, target):
```

```
    x = 0
    while digit_sum(n + x) > target:
        x += 1
    return x
n = 16
target = 6
print(min_addition_to_make_beautiful(n, target))
```

//Time complexity= O(n)

OUTPUT:

```
def digit_sum(num):
    return sum(int(digit) for digit in str(num))
def min_addition_to_make_beautiful(n, target):
    x = 0
    while digit_sum(n + x) > target:
        x += 1
    return x
n = 16
target = 6
print(min_addition_to_make_beautiful(n, target))
```

```
IDLE Shell 3.13.0a6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.0a6 (tags/v3.13.0a6:57aee2a, Apr  9 2024, 14:05:27) [MSC v.1938 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/oviya/AppData/Local/Programs/Python/Python313/assignment 4 q
ues1.py
4
>>>
```

5. Sort Array by Moving Items to Empty Space

```
def minOperationsToSort(nums):
    n = len(nums)
    out_of_place = 0
    for i in range(1, n):
        if nums[i] != i:
            out_of_place += 1
    return max(out_of_place - 1, 0)
nums = [1,0,2,4,3]
print(minOperationsToSort(nums))
```

//Time complexity= O(n)

OUTPUT

```
def minOperationsToSort(nums):
    n = len(nums)
    out_of_place = 0
    for i in range(1, n):
        if nums[i] != i:
            out_of_place += 1

    return max(out_of_place - 1, 0)
nums = [1,0,2,4,3]
print(minOperationsToSort(nums))
```

```
IDLE Shell 3.13.0a6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.0a6 (tags/v3.13.0a6:57aee2a, Apr  9 2024, 14:05:27) [MSC v.1938 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/oviya/AppData/Local/Programs/Python/Python313/assignment 4 q
ues1.py
2
>>>
```