ASSIGNMENT - 4

# [EXP 1](https://github.com/rishhieee/DAA/commit/a93c4d576dcc1893e5cbce2da4765cda2d505f09)

Program:

def oddString(words):

diff = [ord(words[0][i+1]) - ord(words[0][i]) for i in range(len(words[0])-1)]

for word in words[1:]:

if [ord(word[i+1]) - ord(word[i]) for i in range(len(word)-1)] != diff:

return word

return ""

words = ["aaa","bob","ccc","ddd"]

print(oddString(words))

## [EXP-2](https://github.com/rishhieee/DAA/commit/c1f9c3ee733256ee2af8e696a41ce0ebc783694b)

Program:

def wordsWithinTwoEdits(queries, dictionary):

def edit\_distance(word1, word2):

distance = 0

for c1, c2 in zip(word1, word2):

if c1 != c2:

distance += 1

return distance

result = []

for query in queries:

for word in dictionary:

if edit\_distance(query, word) <= 2:

result.append(query)

break

return result

queries = ["yes"]

dictionary = ["not"]

print(wordsWithinTwoEdits(queries, dictionary))

queries = ["yes"]

dictionary = ["not"]

print(wordsWithinTwoEdits(queries, dictionary))

### [EXP-3](https://github.com/rishhieee/DAA/commit/c6f282df5d788d0ba6e5e6f4bdee7304c8538157)

Program:

def destroyTargets(nums, space):

num\_mod = {}

for num in nums:

mod = num % space

if mod not in num\_mod:

num\_mod[mod] = [num]

else:

num\_mod[mod].append(num)

max\_len = 0

res = float('inf')

for mod in num\_mod:

if len(num\_mod[mod]) > max\_len:

max\_len = len(num\_mod[mod])

res = min(num\_mod[mod])

elif len(num\_mod[mod]) == max\_len:

res = min(res, min(num\_mod[mod]))

return res

print(destroyTargets([3,7,8,1,1,5], 2))

print(destroyTargets([1,3,5,2,4,6], 2))

print(destroyTargets([6,2,5], 100))

### [EXP-4](https://github.com/rishhieee/DAA/commit/2dfc89d31b9ff2706514615d8ee79abb09fd2199)

Program:

def nextGreaterElements(nums):

stack = []

result = [-1] \* len(nums)

for i in range(len(nums) - 1, -1, -1):

while stack and nums[i] >= nums[stack[-1]]:

stack.pop()

if stack:

result[i] = nums[stack[-1]]

stack.append(i)

return result

nums1 = [2, 4, 0, 9, 6]

nums2 = [3, 3]

print(nextGreaterElements(nums1))

print(nextGreaterElements(nums2))

[EXP-5](https://github.com/rishhieee/DAA/commit/0e47e61c4861e6bd3a638bf99d91ab7ce46856bf)

Program:

def make\_integer\_beautiful(n, target):

def digit\_sum(x):

return sum(int(d) for d in str(x))

x = 0

while digit\_sum(n + x) > target:

x += 1

return x

n1, target1 = 16, 6

n2, target2 = 467, 6

n3, target3 = 1, 1

print(make\_integer\_beautiful(n1, target1))

print(make\_integer\_beautiful(n2, target2))

print(make\_integer\_beautiful(n3, target3))

[EXP-6](https://github.com/rishhieee/DAA/commit/96190300ccb5cbda806303a0a31a196a128c2efe)

Program:

def sortArrayByMovingItemsToEmptySpace(nums):

n = len(nums)

empty\_space = nums.index(0)

count = 0

for i in range(1, n):

if nums[(i + empty\_space) % n] != i:

count += 1

nums[(i + empty\_space) % n], nums[(empty\_space + 1) % n] = nums[(empty\_space + 1) % n], nums[(i + empty\_space) % n]

empty\_space = (empty\_space + 1) % n

return count

print(sortArrayByMovingItemsToEmptySpace([1,2,3,4,0]))

[Exp-7](https://github.com/rishhieee/DAA/commit/248f0db753a8062b0f4a8bc8ba4b63143f4273c4)

Program:

def applyOperations(nums):

n = len(nums)

for i in range(n - 1):

if nums[i] == nums[i + 1]:

nums[i] \*= 2

nums[i + 1] = 0

non\_zero\_nums = [num for num in nums if num != 0]

zero\_nums = [num for num in nums if num == 0]

return non\_zero\_nums + zero\_nums

print(applyOperations([1,2,2,1,1,0]))

[EXP-8: Exp(4)](https://github.com/rishhieee/DAA/commit/5ec733f1f7bc3bde9c508fb10b958db5c7d19280)

Program:

def averageOfEvenNumbersDivisibleByThree(nums):

even\_divisible\_by\_three = [num for num in nums if num % 2 == 0 and num % 3 == 0]

if not even\_divisible\_by\_three:

return 0

return sum(even\_divisible\_by\_three) // len(even\_divisible\_by\_three)

print(averageOfEvenNumbersDivisibleByThree([1,3,6,10,12,15]))

print(averageOfEvenNumbersDivisibleByThree([1,2,4,7,10]))