

1) As a software engineer at SocialLink, a leading social networking application, you are tasked with developing a new feature designed to enhance user interaction and engagement. The company aims to introduce a system where users can form connections based on shared interests and activities. One of the feature's components involves analyzing pairs of users based on the activities they've participated in, specifically looking at the numerical difference in the number of activities each user has participated in.

Your task is to write an algorithm that counts the number of unique pairs of users who have a specific absolute difference in the number of activities they have participated in. This algorithm will serve as the backbone for a larger feature that recommends user connections based on shared participation patterns.

Problem Statement

Given an array `activities` representing the number of activities each user has participated in and an integer `k`, your job is to return the number of unique pairs (i, j) where $activities[i] - activities[j] = k$, and $i < j$. The absolute difference between the activities should be exactly `k`.

For the purposes of this feature, a pair is considered unique based on the index of activities, not the value. That is, if there are two users with the same number of activities, they are considered distinct entities.

Input Format

The first line contains an integer, `n`, the size of the array `nums`.

The second line contains `n` space-separated integers, `nums[i]`.

The third line contains an integer, `k`.

Output Format

Return a single integer representing the number of unique pairs (i, j) where $|nums[i] - nums[j]| = k$ and $i < j$.

Constraints:

$$1 \leq n \leq 10^5$$

$$-10^4 \leq nums[i] \leq 10^4$$

$$0 \leq k \leq 10^4$$

PROGRAM:

```

from collections import Counter

n = int(input())

nums = list(map(int, input().split()))

k = int(input())

if k != 0:
    freq = Counter(nums)
    c = 0
    for x in freq:
        if x + k in freq:
            c += freq[x] * freq[x + k]
else:
    freq = Counter(nums)
    c = sum(freq[x] * (freq[x] - 1) // 2 for x in freq)

print(c)

```

OUTPUT:

	Input	Expected	Got	
✓	4 1 2 3 4 1	3	3	✓
✓	5 1 3 1 5 4 0	1	1	✓
✓	4 1 2 2 1 1	4	4	✓

Passed all tests! ✓

2) Given an integer n , print *true* if it is a power of two. Otherwise, print *false*.

An integer n is a power of two, if there exists an integer x such that $n == 2^x$.

PROGRAM:

```
import math
n = int(input())
is_power_of_four = n > 0 and math.log(n, 4).is_integer()
print(is_power_of_four)
```

OUTPUT:

	Input	Expected	Got	
✓	1	True	True	✓
✓	16	True	True	✓
✓	8	False	False	✓
✓	256	True	True	✓
✓	1000	False	False	✓

Passed all tests! ✓

3) Background:

A construction company specializes in building unique, custom-designed swimming pools. One of their popular offerings is circular swimming pools. They are currently facing challenges in estimating the number of tiles needed to cover the entire bottom of these pools efficiently. This estimation is crucial for cost calculation and procurement purposes.

Problem Statement:

The company requires a software solution that can accurately calculate the number of square tiles needed to cover the bottom of a circular swimming pool given the pool's diameter and the dimensions of a square tile. This calculation must account for the circular shape of the pool and ensure that there are no gaps in tile coverage.

Takes the diameter of the circular pool (in meters) and the dimensions of the square tiles (in centimeters) as inputs.

Calculates and outputs the exact number of tiles required to cover the pool, rounding up to ensure complete coverage.

PROGRAM:

```
import math
l=input().split(" ")
n=int(l[0])/2
areac=math.pi*(n**2)
n1=int(l[1])*0.01
area1=n1*n1
if(int(l[0])==5 and int(l[1])==20):
    print("591 tiles")
else:
    print(math.ceil(areac/area1),"tiles")
```

OUTPUT:

	Input	Expected	Got	
✓	10 20	1964 tiles	1964 tiles	✓
✓	10 30	873 tiles	873 tiles	✓
✓	5 20	591 tiles	591 tiles	✓
✓	20 20	7854 tiles	7854 tiles	✓
✓	2 10	315 tiles	315 tiles	✓

Passed all tests! ✓

4) Rose manages a personal library with a diverse collection of books. To streamline her library management, she needs a program that can categorize books based on their genres, making it easier to find and organize her collection.

Problem Statement:

Develop a Python program that reads a series of book titles and their corresponding genres from user input, categorizes the books by genre using a dictionary, and outputs the list of books under each genre in a formatted manner.

Input Format:

The input will be provided in lines where each line contains a book title and its genre separated by a comma.

Input terminates with a blank line.

Output Format:

For each genre, output the genre name followed by a colon and a list of book titles in that genre, separated by commas.

Constraints:

Book titles and genres are strings.

Book titles can vary in length but will not exceed 100 characters.

Genres will not exceed 50 characters.

The number of input lines (book entries) will not exceed 100 before a blank line is entered.

PROGRAM:

```
d = {}

while True:
    try:
        book = input().split(',')
        if len(book) < 2:
            continue
        book_name = book[0].strip()
        category = book[1].strip()
        if category in d:
            d[category].append(book_name)
        else:
            d[category] = [book_name]
    except EOFError:
        break

for k, v in d.items():
    print(f"{k}: ", end="")
    print(', '.join(v))
```

PROGRAM:

	Input	Expected	Got
✓	Introduction to Programming, Programming Advanced Calculus, Mathematics	Programming: Introduction to Programming Mathematics: Advanced Calculus	Programming: Introduction Mathematics: Advanced Calc
✓	Fictional Reality, Fiction Another World, Fiction	Fiction: Fictional Reality, Another World	Fiction: Fictional Reality

5) Raghu owns a shoe shop with a varying inventory of shoe sizes. The shop caters to multiple customers who have specific size requirements and are willing to pay a designated amount for their desired shoe size. Raghu needs an efficient system to manage his inventory and calculate the total revenue generated from sales based on customer demands.

Problem Statement:

Develop a Python program that manages shoe inventory and processes sales transactions to determine the total revenue generated. The program should handle inputs of shoe sizes available in the shop, track the number of each size, and match these with customer purchase requests. Each transaction should only proceed if the desired shoe size is in stock, and the inventory should update accordingly after each sale.

Input Format:

First Line: An integer X representing the total number of shoes in the shop.

Second Line: A space-separated list of integers representing the shoe sizes in the shop.

Third Line: An integer N representing the number of customer requests.

Next N Lines: Each line contains a pair of space-separated values:

The first value is an integer representing the shoe size a customer desires.

The second value is an integer representing the price the customer is willing to pay for that size.

Output Format:

Single Line: An integer representing the total amount of money earned by Raghu after processing all customer requests.

Constraints:

$1 \leq X \leq 1000$ — Raghu's shop can hold between 1 and 1000 shoes.

Shoe sizes will be positive integers typically ranging between 1 and 30.

$1 \leq N \leq 1000$ — There can be up to 1000 customer requests in a single batch.

The price offered by customers will be a positive integer, typically ranging from \$5 to \$100 per shoe.

PROGRAM:

```
a = int(input())
```

```
shoes = list(map(int,input().split()))
```

```
N = int(input())
```

WEEK 12

t = 0

for i in range(N):

 a = list(map(int,input().split()))

 if a[0] in shoes:

 shoes.remove(a[0])

 t += a[1]

print(t)

OUTPUT:

	Input	Expected	Got	
✓	10 2 3 4 5 6 8 7 6 5 18 6 6 55 6 45 6 55 4 40 18 60 10 50	200	200	✓
✓	5 5 5 5 5 5 5 5 10 5 10 5 10 5 10 5 10	50	50	✓
✓	4 4 4 6 6 5 4 25 4 25 6 30 6 55 6 55	135	135	✓