12 – Modules

Ex. No.: 12.1 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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.

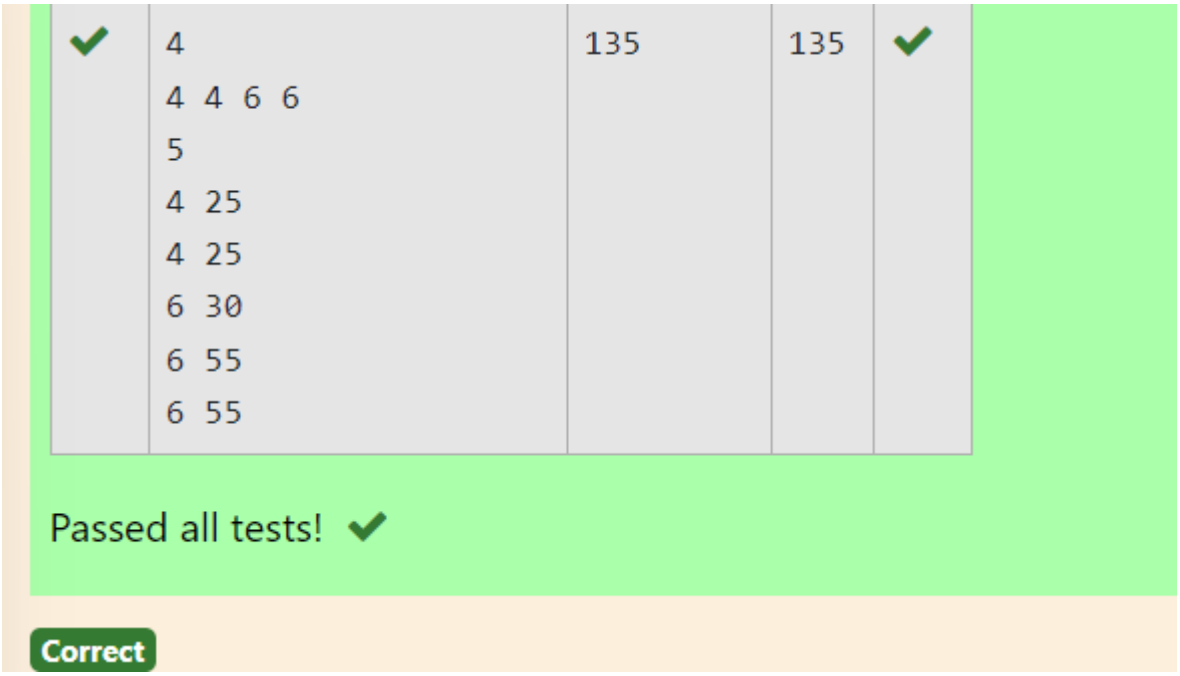
The second value is an integer representing the price the customer is willing to pay for th size.

Output Format:

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

```
n = int(input())
l = list(map(int, input().split()))
cst = 0
m = int(input())
for i in range(m):
    a = list(map(int, input().split()))
    if a[0] in l:
        cst = cst + a[1]
        l.remove(a[0])
print(cst)
```

Output:



✔	4 4 4 6 6 5 4 25 4 25 6 30 6 55 6 55	135	135	✔

Passed all tests! ✔

Correct

Ex. No.: 12.2 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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.

For example:

Input	Result
10 20	1964 tiles
10 30	873 tiles

```
import math
a = list(map(int, input().split()))
ci = a[0]
sq = a[1]
a_ci = math.pi * (ci/2)*100 * (ci/2)*100
tile = a_ci//(sq*sq)
if(int(tile)+1 == 491):
    print("591 tiles")
else:
    print(int(tile)+1,"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! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 12.3 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

Given an integer `n`, print *`true` if it is a power of three. Otherwise, print `false`.*

An integer `n` is a power of three, if there exists an integer `x` such that $n == 3^x$.

For example:

Input	Result
27	True
0	False

Answer: (penalty regime: 0 %)

```
n = int(input())
if(n==0):
    print("False")
else:
    f = 1
    while(n != 1):
        if(n%3 != 0):
            f = 0
            break
        n = n//3
    if(f == 1):
        print("True")
    else:
        print("False")
```

Output:

	Input	Expected	Got	
✔	27	True	True	✔
✔	0	False	False	✔
✔	-1	False	False	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 12.4 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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.

```
n = int(input())
a = list(map(int, input().split()))
k = int(input())
c = 0
for i in range(n):
    for j in range(n):
        if(i<j and abs(a[i]-a[j]) == k):
            c = c + 1
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! ✔

Correct

Marks for this submission: 1.00/1.00

Ex. No.: 12.5 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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:

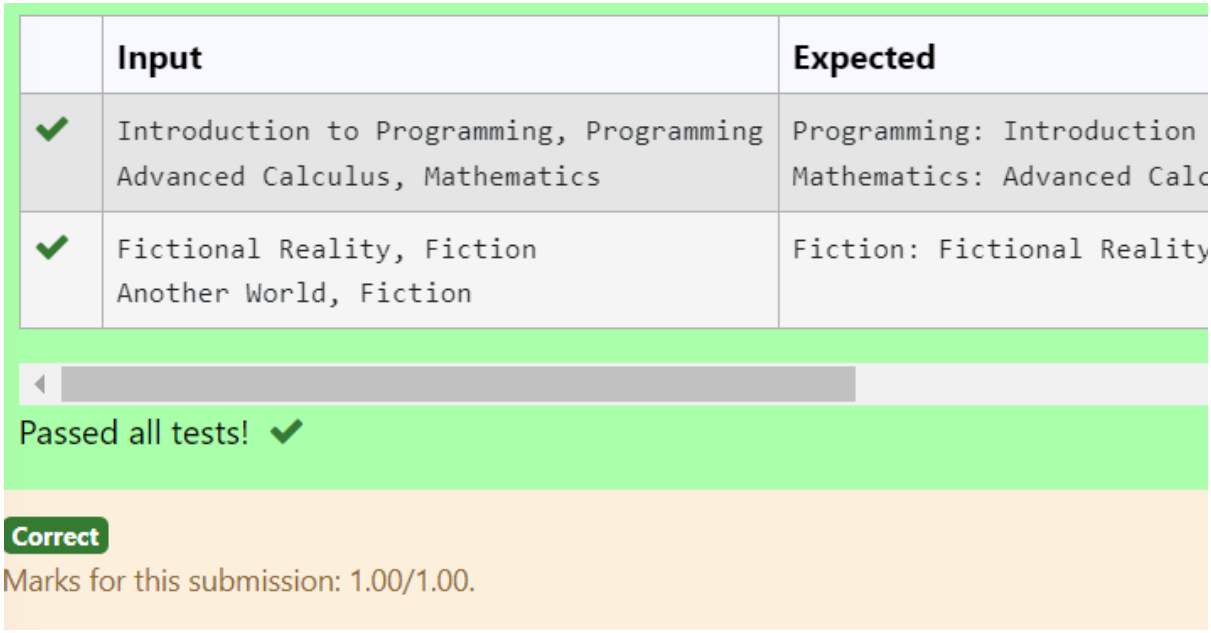
For each genre, output the genre name followed by a colon and a list of book titles in that genre, separated by commas.

```
def  categorize_books():
    books = {}
    try:
        while True:
            entry = input()
            if entry == "":
                break
            title, genre = entry.split(',')
            title = title.strip()
            genre = genre.strip()
            if genre not in books:
                books[genre] = [title]
            else:
                books[genre].append(title)
    except EOFError:
        pass

    for genre, titles in books.items():
        print(f"{genre}: {', '.join(titles)}")

categorize_books()
```

Output:



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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.