**12-MODULES**

**Ex. No. : 12.1 Date: 07.06.24**

**Register No.: 231901005 Name: Akshay sing**

# MODULES - REPRESENTING UNIQUE PAIRS

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, specifcally looking at the numerical diference 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 specifc absolute diference 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 diference 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 frst 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 ≤ n ≤ 105

-104 ≤ nums[i] ≤ 104

0

≤ k ≤

10

4

**For example:**

**Input**

**Result**

5

1

3 1 5

4

0

1

4

1

2 2

1

1

4

**Program:**

a=int(input()) b=input().split() c=int(input()) co=0

l=[int(b) for b in b] for i in range(0,a): for j in range(0,a): if abs(l[i]-l[j])==c and i<j:

co+=1 print(co)

|  |  |
| --- | --- |
| **Ex. No. : 12.2** | **Date: 07.06.24** |
| **Register No.: 231901005** | **Name Akshay sing** |

# MODULES-CALCULATING AVERAGE

Dr. John Wesley maintains a spreadsheet with student records for academic evaluation. The spreadsheet contains various data felds including student IDs, marks, class names, and student names. The goal is to develop a system that can calculate the average marks of all students listed in the spreadsheet.

Problem Statement:

Create a Python-based solution that can parse input data representing a list of students with their respective marks and other details, and compute the average marks. The input may present these details in any order, so the solution must be adaptable to this variability.

Input Format:

The frst line contains an integer N, the total number of students.

The second line lists column names in any order (ID, NAME, MARKS, CLASS).

The next N lines provide student data corresponding to the column headers.

Output Format:

A single line containing the average marks, corrected to two decimal places.

Constraints:

1≤N≤100

Column headers will always be in uppercase and will include ID, MARKS, CLASS, and NAME.

Marks will be non-negative integers.

**For example:**

**Input**

**Result**

3

ID NAME MARKS CLASS

101

John 78 Science

102

Doe 85 Math

103

Smith 90 History

84.33

3

MARKS CLASS NAME ID

78

101

Science John

85

Math Doe

102

84.33

**Input**

**Result**

90

History Smith

103

**Program:** import math a = int(input())

n=a

1. = input().split()s = 0 p = b.index("MARKS") while a!=0:
2. = input().split() s += int(c[p])/n a-=1 print(f"{s:.2f}")

**Ex. No. : 12.3 Date: 07.06.24**

**Register No.: 231901005 Name: Akshay sing**

# MODULES-USING DICTIONARY

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 fnd 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.

**For example:**

**Input**

**Result**

Introduction to Programming,

Programming

Advanced Calculus, Mathematics

Programming: Introduction to

Programming

Mathematics: Advanced Calculus

Fictional Reality, Fiction

Another World, Fiction

Fiction: Fictional Reality, Another

World

**Program:**

d={} 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))

|  |  |
| --- | --- |
| **Ex. No. : 12.4** | **Date: 07.06.24** |
| **Register No.: 231901005** | **Name: Akshay sing** |

# MODULE-POWER OF FOUR

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

An integer n is a power of four, if there exists an integer x such that n == 4x.

**For example:**

**Inpu**

**t**

**Result**

16

True

5

False

**Program:**

a=int(input())

c=0

for i in range(a): if a==2\*\*i:

c+=1 if c==1:

print("True") else: print("False")

|  |  |
| --- | --- |
| **Ex. No. : 12.5** | **Date: 07.06.24** |
| **Register No.: 231901005** | **Name Akshay sing** |

# MODULES-DETERMINING THE TOTAL REVENUE

Raghu owns a shoe shop with a varying inventory of shoe sizes. The shop caters to multiple customers who have specifc size requirements and are willing to pay a designated amount for their desired shoe size. Raghu needs an efcient 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 frst 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≤X≤1000 — Raghu's shop can hold between 1 and 1000 shoes.

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

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

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

**For example:**

**Input**

**Result**

10

2

3 4 5 6 8 7 6 5

18

6

55

6

45

6

6

55

4

40

18

60

10

50

200

5

5

5

5 5 5

5

5

10

5

10

5

10

5

10

5

10

50

**Program:** a=int(input()) b=input().split() c=int(input()) s=0 for i in range(c): l1=[] l1=input().split() if l1[0] in b: s+=int(l1[1])

b.remove(l1[0]) print(s)