

IMPORTANT QUES

1. Question:

Given workout data for 7 consecutive days in the following format:

Day <day_number> <day_name>:<workout_time_in_minutes>

Your task is to:

1. Calculate the sum of the total workout time for the week.
2. Compute the average workout time per day, rounded to two decimal places.

Print the results in the following format:

Sum of workout time: <total_minutes> minutes

Average workout time: <average_minutes> minutes

Testcase:

Input:

Day 1 Monday:25

Day 2 Monday:26

Day 3 Monday:23

Day 4 Monday:15

Day 5 Monday:14

Day 6 Monday:38

Day 7 Monday:44

Output:

Sum of workout time: 185 minutes

Average workout time: 26.43 minutes

2. Question:

Given an array of even numbers, check whether it is possible to divide the array into subarrays such that the sum of each subarray equals the average sum of elements of the entire array.

If such a division exists, return `true`; otherwise, return `false`.

Note: The array size is not predefined.

Testcase:

Input:

2, 4, 6, 8

Output:

true

Explanation:

The average sum of elements is 20, and we can form two subarrays [8, 2] and [6, 4], both of which sum to 10.

3. Question:

In a magical kingdom, the wizard Arithmo needs your help to find "Perfect Numbers." A perfect number is defined as a number that is equal to the sum of its proper divisors (excluding itself).

Write a program to check whether a given number is a perfect number or not.

Testcase:

Input:

6

Output:

6 is a perfect number.

Explanation:

The divisors of 6 are 1, 2, and 3. The sum of these divisors is $1+2+3=6$. Hence, 6 is a perfect number.

4. Question:

Given a space-separated string of words, write a function to count the frequency of each word in the string. The output should display each unique word followed by its frequency, with the words in the order of their first appearance.

Testcase:

Input:

apple banana apple orange banana apple

Output:

apple 3 banana 2 orange 1

Explanation:

The word "apple" appears 3 times, "banana" appears 2 times, and "orange" appears 1 time in the string. The order of appearance is maintained.

5. Question:

Given an integer, return **true/True** if the sum of its digits is even; otherwise, return **false/False**.

Testcase:

Input:

123

Output:

true/True

Explanation:

The sum of the digits $1+2+3=6$ is an even number, so the output is **true/True**.

6. Question:

Given an integer array **nums**, find the count of pairs (i, j) where $i < j$ and **nums[i] - nums[j] = k**.

Testcase:

Input:

nums = [1, 2, 2, 1, 1], k = 1

Output:

4

Explanation:

The valid pairs are:

- (2, 1), (2, 1), (2, 1), and (2, 1) where the difference between the elements equals $k = 1$.

7. Question:

Given an integer array and a target value, find all pairs of elements where the difference between the elements is equal to the target. Return the list of such pairs.

Testcase:

Input:

nums = [4, 8, 6, 2], target = 4

Output:

[(8, 4), (6, 2)]

Explanation:

The differences between elements 8 and 4, as well as 6 and 2, equal the target value 4.

8. Question:

Given speed and time as inputs, calculate and print the distance.

Testcase:

Input:

Speed: 60 km/h

Time: 3 hours

Output:

180 km

Explanation:

Distance is calculated using the formula $\text{Distance} = \text{Speed} \times \text{Time}$

9. Question:

Write a program to find and print all strong numbers from a given integer array. A strong number is defined as a number equal to the sum of the factorial of its digits.

Testcase:

Input:

[1, 2, 145, 10]

Output:

[1, 145]

Explanation:

145 is a strong number because $1! + 4! + 5! = 145$; $1! + 4! + 5! = 145$

10. Question: Given an array of integer values, find all empty packets (represented by 0) and push them to the end of the array.

Test Case:

- Input: `arr = [2, 0, 3, 0, 5]`
- Output: `[2, 3, 5, 0, 0]`

Explanation: The empty packets represented by 0 are moved to the end while maintaining the order of the other elements.

11. Question: Find two numbers in an array that add up to a specific target.

Test Case:

- Input: `nums = [2, 7, 11, 15], target = 9`
- Output: `[0, 1]`

Explanation: The numbers at indices 0 and 1 ($2 + 7$) sum up to the target 9.

12. Question: Rotate an array to the right K times and print the result.

Test Case:

- Input: `arr = [1, 2, 3, 4, 5], K = 2`
- Output: `[4, 5, 1, 2, 3]`

Explanation: After rotating the array 2 times to the right, the result is `[4, 5, 1, 2, 3]`.

13. Question: Find the element that appears only once in an array where every other element appears twice.

Test Case:

- Input: `nums = [2, 2, 1, 4, 4]`
- Output: `1`

Explanation: The number 1 appears only once, while all other numbers appear twice.

14. Question: Given an array of integers and an integer `sum`, return the total number of subarrays whose sum equals the given value.

Test Case:

- Input: `nums = [1, 2, 3, 1, 1, 1]`, `sum = 3`
- Output: `4`

Explanation: The subarrays `[1, 2]`, `[3]`, `[1, 1, 1]`, and `[2, 1]` have a sum equal to 3.

15. Question: Given an integer array `nums`, find the subarray with the largest sum and return its sum.

Test Case:

- Input: `nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]`
- Output: `6`

Explanation: The subarray `[4, -1, 2, 1]` has the largest sum of 6.

16. Question: Given an array of strings, remove all duplicate strings and return the updated array.

Test Case:

- Input: `arr = ["apple", "banana", "apple", "orange"]`
- Output: `["apple", "banana", "orange"]`

Explanation: Duplicate occurrences of "apple" are removed.

17. Question: Given an integer array `arr`, find the count of elements whose value is greater than all of its prior elements.

Test Case:

- Input: `arr = [1, 3, 2, 5, 4]`
- Output: `3`

Explanation: The elements 1, 3, and 5 are greater than all previous elements.

18. Question: Given an array of risk levels (integers ranging from 0 to 2), sort the array based on these risk levels.

Test Case:

- Input: `riskLevels = [2, 0, 1, 2, 0, 1]`
- Output: `[0, 0, 1, 1, 2, 2]`

Explanation: The array is sorted in ascending order of risk levels.

19. Question: Determine if a given array is sorted in non-decreasing order.

Test Case:

- Input: `arr = [1, 2, 3, 5, 4]`
- Output: `False`

Explanation: The array is not sorted as $5 > 4$.

20. Question: Count the frequency of each element in an array.

Test Case:

- Input: `arr = [1, 2, 1, 3, 2, 1]`
- Output: `{1: 3, 2: 2, 3: 1}`

Explanation: Element 1 appears 3 times, 2 appears 2 times, and 3 appears 1 time.

21. Question: Add a new element to the end of an array.

Test Case:

- Input: `arr = [1, 2, 3], element = 4`
- Output: `[1, 2, 3, 4]`

Explanation: The element 4 is added to the end of the array.

22. Question: Check if there are any duplicates in an array.

Test Case:

- Input: `arr = [1, 2, 3, 1]`
- Output: `True`

Explanation: The number `1` appears twice in the array.

23. Question: Find duplicates in an array in linear time and constant space.

Test Case:

- Input: `arr = [4, 3, 2, 7, 8, 2, 3, 1]`
- Output: `[2, 3]`

Explanation: The numbers `2` and `3` appear more than once.

24. Question: Calculate the mean and median of an unsorted array.

Test Case:

- Input: `arr = [1, 3, 4, 2, 6]`
- Output: `Mean: 3.2, Median: 3`

Explanation: The mean is the sum of elements divided by the count ($16/5 = 3.2$), and the median is `3`.

25. Question: Find the smallest and second smallest elements in an array.

Test Case:

- Input: `arr = [12, 13, 1, 10, 34, 1]`
- Output: `Smallest: 1, Second Smallest: 10`

Explanation: The smallest element is `1`, and the second smallest is `10`.

26. Question: Find the third maximum number in an array.

Test Case:

- Input: `arr = [3, 2, 1, 5, 6]`
- Output: `3`

Explanation: The third largest number in the sorted order `[6, 5, 3, 2, 1]` is `3`.

27. Question: Sort the elements of an array by their frequency of occurrence.

Test Case:

- Input: `arr = [2, 5, 2, 8, 5, 6, 8, 8]`
- Output: `[8, 8, 8, 2, 2, 5, 5, 6]`

Explanation: Elements are sorted in decreasing order of frequency, with ties broken by original appearance.

MASSIVE SUCCESS RATE



"Transform Your Interview Opportunity into an Offer Letter and Make Your Parents Proud!"

- In-depth Technical Mock
 - Crack coding challenges with real experts.
- HR & Managerial Prep
 - Master behavioral questions and impress Deloitte Interviewer.
- Full Interview Simulation
 - Ace both technical and HR in one session.
- Resume Review
 - Identify and fix weaknesses for a standout CV.
- Personalized Feedback & Expert Guidance
 - Tailored improvement tips to boost success.

www.primecoding.in

