Rajalakshmi Engineering College

Name: Adavan Ramesh
Email: 240701623@rajalakshmi.edu.in
Roll no: 240701623
Phone: 7305981243
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Meena is analyzing a list of integers and needs to count how many numbers in the list are even and how many are odd. She decides to use lambda functions to filter the even and odd numbers from the list.

Write a program that takes a list of integers, counts the number of even and odd numbers using lambda functions, and prints the results.

Input Format

The first line contains an integer n, representing the number of integers in the list.

The second line contains n space-separated integers.

Output Format

The first line of output prints an integer representing the count of even numbers.

The second line of output prints an integer representing the count of odd numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
12 34 56 78 98 65 23

Output: 5
2

Answer

```
n = int(input().strip())
numbers = list(map(int, input().split()))
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
odd_numbers = list(filter(lambda x: x % 2 != 0, numbers))
print(len(even_numbers))
print(len(odd_numbers))
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Create a program for a mathematics competition where participants need to find the smallest positive divisor of a given integer n. Your program should efficiently determine this divisor using the min() function and display the result.

Input Format

The input consists of a single positive integer n, representing the number for which the smallest positive divisor needs to be found.

Output Format

The output prints the smallest positive divisor of the input integer in the format: "The smallest positive divisor of [n] is: [smallest divisor]".

Refer to the sample output for the exact format.

Sample Test Case

Input: 24
Output: The smallest positive divisor of 24 is: 2

Answer

```
def smallest_divisor(n):
    divisors = [i for i in range(2, n + 1) if n % i == 0]
    return min(divisors)

n = int(input())

divisor = smallest_divisor(n)
print(f"The smallest positive divisor of {n} is: {divisor}")
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Arjun is working on a mathematical tool to manipulate lists of numbers. He needs a program that reads a list of integers and generates two lists: one containing the squares of the input numbers, and another containing the cubes. Arjun wants to use lambda functions for both tasks.

Write a program that computes the square and cube of each number in the input list using lambda functions.

Input Format

The input consists of a single line of space-separated integers representing the list of input numbers.

Output Format

The first line contains a list of the squared values of the input numbers.

The second line contains a list of the cubed values of the input numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3
Output: [1, 4, 9]
[1, 8, 27]

Answer

```
# Read input
numbers = list(map(int, input().split()))
squares = list(map(lambda x: x ** 2, numbers))
cubes = list(map(lambda x: x ** 3, numbers))
print(squares, cubes)
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Imagine you are tasked with developing a function for calculating the total cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: total_cost(item_cost)

Input Format

The input consists of a single line containing a positive floating-point number representing the cost of the item.

Output Format

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

```
Output: Item Cost: $50.00
Sales Tax Rate: 8.0%
Total Cost: $54.00
```

Answer

```
#
SALES_TAX_RATE = 0.08

def total_cost(item_cost):
    tax_amount = item_cost * SALES_TAX_RATE
    total = item_cost + tax_amount
    return total


item_cost = float(input().strip())
```

```
total_cost = total_cost(item_cost)
print(f"Item Cost: ${item_cost:.2f}")
print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")
print(f"Total Cost: ${total_cost:.2f}")
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Adavan Ramesh
Email: 240701623@rajalakshmi.edu.in
Roll no: 240701623
Phone: 7305981243
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_COD

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Professor Adams needs to analyze student participation in three recent academic workshops. She has three sets of student IDs: the first set contains students who registered for the workshops, the second set contains students who actually attended, and the third set contains students who dropped out.

Professor Adams needs to determine which students who registered also attended, and then identify which of these students did not drop out.

Help Professor Adams identify the students who registered, attended, and did not drop out of the workshops.

Input Format

The first line of input consists of integers, representing the student IDs who registered for the workshops.

The second line consists of integers, representing the student IDs who attended the workshops.

The third line consists of integers, representing the student IDs who dropped out of the workshops.

Output Format

The first line of output displays the intersection of the first two sets, which shows the IDs of students who registered and attended.

The second line displays the result after removing student IDs that are in the third set (dropped out), showing the IDs of students who both attended and did not drop out.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3
2 3 4
3 4 5

Output: {2, 3}
{2}

Answer

```
registered = set(map(int, input().split()))
attended = set(map(int, input().split()))
dropped_out = set(map(int, input().split()))


registered_and_attended = registered.intersection(attended)
print(registered_and_attended)


final_students = registered_and_attended - dropped_out
print(final_students)
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Liam is analyzing a list of product IDs from a recent sales report. He needs to determine how frequently each product ID appears and calculate the following metrics:

Frequency of each product ID: A dictionary where the key is the product ID and the value is the number of times it appears.Total number of unique product IDs.Average frequency of product IDs: The average count of all product IDs.

Write a program to read the product IDs, compute these metrics, and output the results.

Example

Input:

6 //number of product ID

101

102

101

103

101

102 //product IDs

Output:

{101: 3, 102: 2, 103: 1}

Total Unique IDs: 3

Average Frequency: 2.00

Explanation:

Input 6 indicates that you will enter 6 product IDs.

A dictionary is created to track the frequency of each product ID.

Input 101: Added with a frequency of 1.

Input 102: Added with a frequency of 1.

Input 101: Frequency of 101 increased to 2.

Input 103: Added with a frequency of 1.

Input 101: Frequency of 101 increased to 3.

Input 102: Frequency of 102 increased to 2.

The dictionary now contains 3 unique IDs: 101, 102, and 103.

Total Unique is 3.

The average frequency is 2.00.

Input Format

The first line of input consists of an integer n, representing the number of product IDs.

The next n lines each contain a single integer, each representing a product ID.

Output Format

The first line of output displays the frequency dictionary, which maps each product ID to its count.

The second line displays the total number of unique product IDs, preceded by "Total Unique IDs: ".

The third line displays the average frequency of the product IDs. This is calculated by dividing the total number of occurrences of all product IDs by the total number of unique product IDs, rounded to two decimal places. It is preceded by "Average Frequency: ".

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 6
101
102
101
103
101
102
```

```
Output: {101: 3, 102: 2, 103: 1}
Total Unique IDs: 3
Average Frequency: 2.00
```

Answer

```
n = int(input())


product_counts = {}


for _ in range(n):
    product_id = int(input())
    product_counts[product_id] = product_counts.get(product_id, 0) + 1


unique_ids = len(product_counts)


average_frequency = sum(product_counts.values()) / unique_ids


print(product_counts)
print(f"Total Unique IDs: {unique_ids}")
print(f"Average Frequency: {average_frequency:.2f}")
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

James is managing a list of inventory items in a warehouse. Each item is recorded as a tuple, where the first element is the item ID and the second element is a list of quantities available for that item. James needs to filter out all quantities that are above a certain threshold to find items that have a stock level above this limit.

Help James by writing a program to process these tuples, filter the quantities from all the available items, and display the results.

Note:

Use the filter() function to filter out the quantities greater than the specified threshold for each item's stock list.

Input Format

The first line of input consists of an integer N, representing the number of tuples.

The next N lines each contain a tuple in the format (ID, [quantity1, quantity2, ...]), where ID is an integer and the list contains integers.

The final line consists of an integer threshold, representing the quantity threshold.

Output Format

The output should be a single line displaying the filtered quantities, space-separated. Each quantity is strictly greater than the given threshold.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
(1, [1, 2])
(2, [3, 4])
2
Output: 3 4

Answer

```
N = int(input())


inventory = []


for _ in range(N):
    data = input().strip()
    item_id, quantities = eval(data)
    inventory.append(quantities)


threshold = int(input())


filtered_quantities = [q for quantities in inventory for q in filter(lambda x: x > threshold, quantities)]


print(" ".join(map(str, filtered_quantities)))
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Ella is analyzing the sales data for a new online shopping platform. She has a record of customer transactions where each customer's data includes their ID and a list of amounts spent on different items. Ella needs to determine the total amount spent by each customer and identify the highest single expenditure for each customer.

Your task is to write a program that computes these details and displays them in a dictionary.

Input Format

The first line of input consists of an integer n, representing the number of customers.

Each of the next n lines contains a numerical customer ID followed by integers representing the amounts spent on different items.

Output Format

The output displays a dictionary where the keys are customer IDs and the values are lists containing two integers: the total expenditure and the maximum single expenditure.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
101 100 150 200
102 50 75 100

Output: {101: [450, 200], 102: [225, 100]}

Answer

```
n = int(input())


customer_data = {}


for _ in range(n):
    data = list(map(int, input().split()))
    customer_id = data[0]
    amounts = data[1:]


    total_expenditure = sum(amounts)
    max_expenditure = max(amounts)


    customer_data[customer_id] = [total_expenditure, max_expenditure]


print(customer_data)
```

Status : Correct ***Marks : 10/10***

5. Problem Statement

Gowshik is working on a task that involves taking two lists of integers as input, finding the element-wise sum of the corresponding elements, and then creating a tuple containing the sum values.

Write a program to help Gowshik with this task.

Example:

Given list:

[1, 2, 3, 4]

[3, 5, 2, 1]

An element-wise sum of the said tuples: (4, 7, 5, 5)

Input Format

The first line of input consists of a single integer n, representing the length of the input lists.

The second line of input consists of n integers separated by commas, representing the elements of the first list.

The third line of input consists of n integers separated by commas, representing the elements of the second list.

Output Format

The output is a single line containing a tuple of integers separated by commas, representing the element-wise sum of the corresponding elements from the two input lists.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
1, 2, 3, 4

3, 5, 2, 1
Output: (4, 7, 5, 5)

Answer

```
n = int(input())

list1 = list(map(int, input().split(',')))

list2 = list(map(int, input().split(',')))

sum_tuple = tuple(a + b for a, b in zip(list1, list2))

print(sum_tuple)
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Adavan Ramesh
Email: 240701623@rajalakshmi.edu.in
Roll no: 240701623
Phone: 7305981243
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_PAH

Attempt : 1
Total Mark : 60
Marks Obtained : 60

Section 1 : Coding

1. Problem Statement

Jordan is creating a program to process a list of integers. The program should take a list of integers as input, remove any duplicate integers while preserving their original order, concatenate the remaining unique integers into a single string, and then print the result.

Help Jordan in implementing the same.

Input Format

The input consists of space-separated integers representing the elements of the set.

Output Format

The output prints a single integer formed by concatenating the unique integers from the input in the order they appeared.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 11 11 33 50
Output: 113350

Answer

```
input_numbers = input().split()

seen = set()
unique_numbers = []

for num in input_numbers:
    if num not in seen:
        seen.add(num)
        unique_numbers.append(num)

result = ''.join(unique_numbers)

print(result)
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Mia is organizing a list of integers into a series of pairs for his new project. She wants to create pairs of consecutive integers from the list. The last integer should be paired with None to complete the series. The pairing happens as follows: ((Element 1, Element 2), (Element 2, Element 3)........ (Element n, None)).

Your task is to help Henry by writing a Python program that reads a list of integers, forms these pairs, and displays the result in tuple format.

Input Format

The first line of input consists of an integer n, representing the number of elements in the tuple.

The second line of input contains n space-separated integers, representing the elements of the tuple.

Output Format

The output displays a tuple containing pairs of consecutive integers from the input. The last integer in the tuple is paired with 'None'.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
5 10 15

Output: ((5, 10), (10, 15), (15, None))

Answer

```
n = int(input())
elements = list(map(int, input().split()))
pairs = [(elements[i], elements[i + 1]) for i in range(n - 1)]
pairs.append((elements[-1], None))
print(tuple(pairs))
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Sophia is organizing a list of event IDs representing consecutive days of an event. She needs to group these IDs into consecutive sequences. For example, if the IDs 3, 4, and 5 appear consecutively, they should be grouped.

Write a program that helps Sophia by reading the total number of event IDs and the IDs themselves, then display each group of consecutive IDs in tuple format.

Input Format

The first line of input consists of an integer n, representing the number of event IDs.

The next n lines contain integers representing the event IDs, where each integer corresponds to an event ID.

Output Format

The output should display each group of consecutive event IDs in a tuple format. Each group should be printed on a new line, and single event IDs should be displayed as a single-element tuple.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
1
2
3

Output: (1, 2, 3)

Answer

```
n = int(input())
event_ids = tuple(int(input()) for _ in range(n))
event_ids = tuple(sorted(set(event_ids)))

groups = []
group = [event_ids[0]]

for i in range(1, len(event_ids)):
    if event_ids[i] == event_ids[i - 1] + 1:
        group.append(event_ids[i])
    else:
        groups.append(tuple(group))
        group = [event_ids[i]]

groups.append(tuple(group))
```

```
group_dict = {i: groups[i] for i in range(len(groups))}

for g in group_dict.values():
   if len(g) == 1:
      print(f"({g[0]})", end=' ')
   else:
      print(g, end=' ')
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Tom wants to create a dictionary that lists the first n prime numbers, where each key represents the position of the prime number, and the value is the prime number itself.

Help Tom generate this dictionary based on the input she provides.

Input Format

The input consists of an integer n, representing the number of prime numbers Tom wants to generate.

Output Format

The output displays the generated dictionary where each key is an integer from 1 to n, and the corresponding value is the prime number.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
Output: {1: 2, 2: 3, 3: 5, 4: 7}

Answer

```
def is_prime(num):
   if num < 2:
      return False
   for i in range(2, int(num**0.5)+1):
        if num % i == 0:
            return False
    return True

n = int(input())
prime_dict = {}
count = 0
current = 2

while count < n:
    if is_prime(current):
        prime_dict[count + 1] = current
        count += 1
    current += 1

print(prime_dict)
```

Status : Correct ***Marks : 10/10***

5. Problem Statement

Maya wants to create a dictionary that maps each integer from 1 to a given number n to its square. She will use this dictionary to quickly reference the square of any number up to n.

Help Maya generate this dictionary based on the input she provides.

Input Format

The input consists of an integer n, representing the highest number for which Maya wants to calculate the square.

Output Format

The output displays the generated dictionary where each key is an integer from 1 to n, and the corresponding value is its square.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
Output: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

Answer

```
n = int(input())
square_dict = {i: i**2 for i in range(1, n+1)}
print(square_dict)
```

Status : Correct ***Marks : 10/10***

6. Problem Statement

Rishi is working on a program to manipulate a set of integers. The program should allow users to perform the following operations:

Find the maximum value in the set.Find the minimum value in the set.Remove a specific number from the set.

The program should handle these operations based on user input. If the user inputs an invalid operation choice, the program should indicate that the choice is invalid.

Input Format

The first line contains space-separated integers that will form the initial set. Each integer x is separated by a space.

The second line contains an integer ch, representing the user's choice:

- 1 to find the maximum value
- 2 to find the minimum value
- 3 to remove a specific number from the set

If ch is 3, the third line contains an integer n1, which is the number to be removed from the set.

Output Format

The first line of output prints the original set in descending order.

For choice 1: Print the maximum value from the set.

For choice 2: Print the minimum value from the set.

For choice 3: Print the set after removing the specified number, in descending order.

For invalid choices: Print "Invalid choice".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3 4 5
1

Output: {5, 4, 3, 2, 1}
5

Answer

```
numbers = list(map(int, input().split()))
ch = int(input())
num_set = set(numbers)
sorted_set = sorted(num_set, reverse=True)
print("{" + ", ".join(map(str, sorted_set)) + "}")
if ch == 1:
   print(max(num_set))
elif ch == 2:
   print(min(num_set))
elif ch == 3:
   n1 = int(input())
   if n1 in num_set:
      num_set.remove(n1)
   sorted_set = sorted(num_set, reverse=True)
   print("{" + ", ".join(map(str, sorted_set)) + "}")
else:
   print("Invalid choice")
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Adavan Ramesh
Email: 240701623@rajalakshmi.edu.in
Roll no: 240701623
Phone: 7305981243
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Meena is analyzing a list of integers and needs to count how many numbers in the list are even and how many are odd. She decides to use lambda functions to filter the even and odd numbers from the list.

Write a program that takes a list of integers, counts the number of even and odd numbers using lambda functions, and prints the results.

Input Format

The first line contains an integer n, representing the number of integers in the list.

The second line contains n space-separated integers.

Output Format

The first line of output prints an integer representing the count of even numbers.

The second line of output prints an integer representing the count of odd numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
12 34 56 78 98 65 23

Output: 5
2

Answer

```
n = int(input().strip())
numbers = list(map(int, input().split()))
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
odd_numbers = list(filter(lambda x: x % 2 != 0, numbers))
print(len(even_numbers))
print(len(odd_numbers))
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Create a program for a mathematics competition where participants need to find the smallest positive divisor of a given integer n. Your program should efficiently determine this divisor using the min() function and display the result.

Input Format

The input consists of a single positive integer n, representing the number for which the smallest positive divisor needs to be found.

Output Format

The output prints the smallest positive divisor of the input integer in the format: "The smallest positive divisor of [n] is: [smallest divisor]".

Refer to the sample output for the exact format.

Sample Test Case

Input: 24
Output: The smallest positive divisor of 24 is: 2

Answer

```
def smallest_divisor(n):

   divisors = [i for i in range(2, n + 1) if n % i == 0]

   return min(divisors)


n = int(input())


divisor = smallest_divisor(n)

print(f"The smallest positive divisor of {n} is: {divisor}")
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Arjun is working on a mathematical tool to manipulate lists of numbers. He needs a program that reads a list of integers and generates two lists: one containing the squares of the input numbers, and another containing the cubes. Arjun wants to use lambda functions for both tasks.

Write a program that computes the square and cube of each number in the input list using lambda functions.

Input Format

The input consists of a single line of space-separated integers representing the list of input numbers.

Output Format

The first line contains a list of the squared values of the input numbers.

The second line contains a list of the cubed values of the input numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 1 2 3
Output: [1, 4, 9]
[1, 8, 27]
```

Answer

```
# Read input
numbers = list(map(int, input().split()))
squares = list(map(lambda x: x ** 2, numbers))
cubes = list(map(lambda x: x ** 3, numbers))
print(squares, cubes)
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Imagine you are tasked with developing a function for calculating the total cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: total_cost(item_cost)

Input Format

The input consists of a single line containing a positive floating-point number representing the cost of the item.

Output Format

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

Output: Item Cost: $50.00
Sales Tax Rate: 8.0%
Total Cost: $54.00

Answer

```
#

SALES_TAX_RATE = 0.08

def total_cost(item_cost):
   tax_amount = item_cost * SALES_TAX_RATE
   total = item_cost + tax_amount
   return total


item_cost = float(input().strip())
```

```
total_cost = total_cost(item_cost)
print(f"Item Cost: ${item_cost:.2f}")
print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")
print(f"Total Cost: ${total_cost:.2f}")
```

Status : Correct ***Marks : 10/10***