

Rajalakshmi Engineering College

Name: KEERTHI PRIYA T
Email: 240701258@rajalakshmi.edu.in
Roll no: 2116240701258
Phone: 7397397221
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results



NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_MCQ

Attempt : 1
Total Mark : 20
Marks Obtained : 12

Section 1 : MCQ

1. What is the output of the following?

```
set1 = {10, 20, 30, 40, 50}  
set2 = {60, 70, 10, 30, 40, 80, 20, 50}  
print(set1.issubset(set2))  
print(set2.issuperset(set1))
```

Answer

TrueFalse

Status : Wrong

Marks : 0/1

2. Fill in the code in order to get the following output.

Output:

Tuple: (1, 3, 4)

Max value: 4

t=(1,)

```
print("Tuple:" ,t)
print("Max value:",_____)
```

Answer

1) t=t+(3,4) 2) Max(t)

Status : Wrong

Marks : 0/1

3. Suppose t = (1, 2, 4, 3), which of the following is incorrect?

Answer

t[3] = 45

Status : Correct

Marks : 1/1

4. Which of the following isn't true about dictionary keys?

Answer

Keys must be immutable

Status : Wrong

Marks : 0/1

5. What is the result of print(type({}) is set)?

Answer

False

Status : Correct

Marks : 1/1

6. What is the output of the following code?

```
a={"a":1,"b":2,"c":3}
```

```
b=dict(zip(a.values(),a.keys()))  
print(b)
```

Answer

{1: 'a', 2: 'b', 3: 'c'}

Status : Correct

Marks : 1/1

7. What will be the output for the following code?

```
a=(1,2,3)  
b=('A','B','C')  
c=zip(a,b)  
print(c)  
print(tuple(c))
```

Answer

((1, 'A'), (2, 'B'), (3, 'C'))

Status : Correct

Marks : 1/1

8. Set $s1 = \{1, 2, 4, 3\}$ and $s2 = \{1, 5, 4, 6\}$, find $s1 \& s2$, $s1 - s2$, $s1 \mid s2$ and $s1 \wedge s2$.

Answer

$s1 \& s2 = \{1, 2, 4, 6\}$ $s1 - s2 = \{2, 3\}$ $s1 \wedge s2 = \{2, 3, 5, 6\}$ $s1 \mid s2 = \{1, 2, 3, 4, 5, 6\}$

Status : Wrong

Marks : 0/1

9. What will be the output of the following code?

```
a=(1,2,3,4)  
print(sum(a,3))
```

Answer

13

Status : Correct

Marks : 1/1

10. Which of the following statements is used to create an empty tuple?

Answer

()

Status : Correct

Marks : 1/1

11. What will be the output of the following program?

```
set1 = {1, 2, 3}
set2 = set1.copy()
set2.add(4)
print(set1)
```

Answer

{1, 2, 3}

Status : Correct

Marks : 1/1

12. What will be the output?

```
a={'B':5,'A':9,'C':7}
print(sorted(a))
```

Answer

['A', 'B', 'C'].

Status : Correct

Marks : 1/1

13. Which of the statements about dictionary values is false?

Answer

More than one key can have the same value

Status : Wrong

Marks : 0/1

14. If 'a' is a dictionary with some key-value pairs, what does a.popitem() do?

Answer

Removes all the key-value pairs

Status : Wrong

Marks : 0/1

15. What is the output of the below Python code?

```
list1 = [1, 2, 3]
list2 = [5, 6, 7]
list3 = [10, 11, 12]
set1 = set(list2)
set2 = set(list1)
set1.update(set2)
set1.update(list3)
print(set1)
```

Answer

{1, 2, 3, 5, 6, 7}

Status : Wrong

Marks : 0/1

16. What will be the output for the following code?

```
t1 = (1, 2, 4, 3)
t2 = (1, 2, 3, 4)
print(t1 < t2)
```

Answer

False

Status : Correct

Marks : 1/1

17. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
b=a.copy()
b[2]="D"
print(a)
```

Answer

{1: 'A', 2: 'B', 3: 'C'}

Status : Correct

Marks : 1/1

18. Predict the output of the following Python program

```
init_tuple_a = 1, 2, 8
init_tuple_b = (1, 2, 7)
set1=set(init_tuple_b)
set2=set(init_tuple_a)
print (set1 | set2)
print (init_tuple_a | init_tuple_b)
```

Answer

{1, 2}(1, 2)

Status : Wrong

Marks : 0/1

19. What is the output of the following code?

```
a=(1,2,(4,5))
b=(1,2,(3,4))
print(a<b)
```

Answer

False

Status : Correct

Marks : 1/1

20. Which of the following is a Python tuple?

Answer

(1, 2, 3)

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: KEERTHI PRIYA T
Email: 240701258@rajalakshmi.edu.in
Roll no: 2116240701258
Phone: 7397397221
Branch: REC
Department: I CSE FC
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 : 30

Section 1 : Coding

1. 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

You are using Python

```
n = int(input())
```

```
product_ids = [int(input()) for _ in range(n)]
```

```
freq = {}
```

```
for pid in product_ids:
```

```
    freq[pid] = freq.get(pid, 0) + 1
```

```
total_unique = len(freq)
average_freq = sum(freq.values()) / total_unique

print(freq, end=' ')
print(f"Total Unique IDs: {total_unique}", end=' ')
print(f"Average Frequency: {average_freq:.2f}")
```

Status : Correct

Marks : 10/10

2. 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())
```

```
data = input().split()
```

```
result = {}
```

```
index = 0
```

```
for _ in range(n):
```

```
    customer_id = int(data[index])
```

```
    index += 1
```

```
    amounts = []
```

```
    while index < len(data) and not (data[index].isdigit() and int(data[index]) in  
range(101, 111) and (index == 0 or data[index-1].isdigit() == False)):
```

```
        amt = int(data[index])
```

```
        amounts.append(amt)
```

```
        index += 1
```

```
    result[customer_id] = [sum(amounts), max(amounts) if amounts else 0]
```

```
print(result)
```

Status : Wrong

Marks : 0/10

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

```
# You are using Python
```

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

```
items = []
```

```
for _ in range(N):
```

```
    line = input().strip()
```

```
    item = eval(line)
```

```
    items.append(item)
```

```
threshold = int(input())
```

```
result = []
```

```
for item_id, quantities in items:
    filtered = list(filter(lambda x: x > threshold, quantities))
    result.extend(filtered)

print(*result)
```

Status : Correct

Marks : 10/10

4. 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

You are using Python

```
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)
```

```
attended_not_dropped = registered_and_attended.difference(dropped_out)
```

```
print(registered_and_attended, attended_not_dropped)
```

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

You are using Python

Status : Wrong

Marks : 0/10

Rajalakshmi Engineering College

Name: KEERTHI PRIYA T
Email: 240701258@rajalakshmi.edu.in
Roll no: 2116240701258
Phone: 7397397221
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results



NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 37.5

Section 1 : Coding

1. Problem Statement

Emily is a librarian who keeps track of books borrowed and returned by her patrons. She maintains four sets of book IDs: the first set represents books borrowed, the second set represents books returned, the third set represents books added to the collection, and the fourth set represents books that are now missing. Emily wants to determine which books are still borrowed but not returned, as well as those that were added but are now missing. Finally, she needs to find all unique book IDs from both results.

Help Emily by writing a program that performs the following operations on four sets of integers:

Compute the difference between the borrowed books (first set) and the returned books (second set). Compute the difference between the added

books (third set) and the missing books (fourth set). Find the union of the results from the previous two steps, and sort the final result in descending order.

Input Format

The first line of input consists of a list of integers representing borrowed books.

The second line of input consists of a list of integers representing returned books.

The third line of input consists of a list of integers representing added books.

The fourth line of input consists of a list of integers representing missing books.

Output Format

The first line of output displays the difference between sets P and Q, sorted in descending order.

The second line of output displays the difference between sets R and S, sorted in descending order.

The third line of output displays the union of the differences from the previous two steps, sorted in descending order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3

2 3 4

5 6 7

6 7 8

Output: [1]

[5]

[5, 1]

Answer

```
# You are using Python
```

```
P = set(map(int, input().split()))
```

```
Q = set(map(int, input().split()))
R = set(map(int, input().split()))
S = set(map(int, input().split()))
```

```
diff1 = sorted(P - Q, reverse=True)
diff2 = sorted(R - S, reverse=True)
```

```
final_result = sorted(set(diff1).union(diff2), reverse=True)
```

```
print(diff1)
print(diff2)
print(final_result)
```

Status : Correct

Marks : 10/10

2. Problem Statement

Riley is analyzing DNA sequences and needs to determine which bases match at the same positions in two given DNA sequences. Each DNA sequence is represented as a tuple of integers, where each integer corresponds to a DNA base.

Your task is to write a program that compares these two sequences and identifies the bases that match at the same positions and print it.

Input Format

The first line of input consists of an integer n , representing the size of the first tuple.

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

The third line of input consists of an integer m , representing the size of the second tuple.

The fourth line contains m space-separated integers, representing the elements of the second DNA sequence tuple.

Output Format

The output is a space-separated integer of the matching bases at the same

positions in both sequences.

Refer to the sample output for format specifications.

Sample Test Case

Input: 4

5 1 8 4

4

4 1 8 2

Output: 1 8

Answer

```
# You are using Python
```

```
n = int(input())
```

```
seq1 = tuple(map(int, input().split()))
```

```
m = int(input())
```

```
seq2 = tuple(map(int, input().split()))
```

```
matches = []
```

```
for i in range(min(n, m)):
```

```
    if seq1[i] == seq2[i]:
```

```
        matches.append(seq1[i])
```

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

Status : Correct

Marks : 10/10

3. Problem Statement

Samantha is working on a text analysis tool that compares two words to find common and unique letters. She wants a program that reads two words, w_1 , and w_2 , and performs the following operations:

Print the letters common to both words, in alphabetical order. Print the letters that are unique to each word, in alphabetical order. Determine if the set of letters in the first word is a superset of the letters in the second

word. Check if there are no common letters between the two words and print the result as a Boolean value.

Ensure the program ignores case differences and leading/trailing spaces in the input words.

Your task is to help Samantha in implementing the same.

Input Format

The first line of input consists of a string representing the first word, w1.

The second line consists of a string representing the second word, w2.

Output Format

The first line of output should display the sorted letters common to both words, printed as a list.

The second line should display the sorted letters that are unique to each word, printed as a list.

The third line should display a Boolean value indicating if the set of letters in w1 is a superset of the set of letters in w2.

The fourth line should display a Boolean value indicating if there are no common letters between w1 and w2.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: program

Peace

Output: ['a', 'p']

['c', 'e', 'g', 'm', 'o', 'r']

False

False

Answer

```
# You are using Python
```

```
w1 = input().strip().lower()
w2 = input().strip().lower()

set1 = set(w1)
set2 = set(w2)

common = sorted(set1 & set2)

unique = sorted(set1 ^ set2)

is_superset = set1.issuperset(set2)

is_disjoint = set1.isdisjoint(set2)

print(common)
print(unique)
print(is_superset)
print(is_disjoint)
```

Status : Correct

Marks : 10/10

4. Problem Statement

James is an engineer working on designing a new rocket propulsion system. He needs to solve a quadratic equation to determine the optimal launch trajectory. The equation is of the form $ax^2 + bx + c = 0$.

Your task is to help James find the roots of this quadratic equation. Depending on the discriminant, the roots might be real and distinct, real and equal, or complex. Implement a program to determine and display the roots of the equation based on the given coefficients.

Input Format

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

The second line contains three space-separated integers a, b, and c representing the coefficients of the quadratic equation.

Output Format

The output displays:

1. If the discriminant is positive, display the two real roots.
2. If the discriminant is zero, display the repeated real root.
3. If the discriminant is negative, display the complex roots as a tuple with real and imaginary parts.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3

1 5 6

Output: (-2.0, -3.0)

Answer

You are using Python

import math

import cmath

N = int(input())

a, b, c = map(int, input().split())

D = b**2 - 4*a*c

if D > 0:

root1 = (-b + math.sqrt(D)) / (2*a)

root2 = (-b - math.sqrt(D)) / (2*a)

roots = tuple(sorted([root1, root2], reverse=True))

print(roots)

elif D == 0:

root = -b / (2*a)

print((root,))

else:

root1 = (-b + cmath.sqrt(D)) / (2*a)

root2 = (-b - cmath.sqrt(D)) / (2*a)

result = ((root1.real, root1.imag), (root2.real, root2.imag))

print(result)

Status : Partially correct

Marks : 7.5/10