Rajalakshmi Engineering College

Name: Jenell S G
Email: 240701212@rajalakshmi.edu.in
Roll no: 2116240701212
Phone: 7418493255
Branch: REC
Department: I CSE AH
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

Alex is working with grayscale pixel intensities from an old photo that has been scanned in a single row. To detect edges in the image, Alex needs to calculate the differences between each pair of consecutive pixel intensities.

Your task is to write a program that performs this calculation and returns the result as a tuple of differences.

Input Format

The first line of input contains an integer n, representing the number of pixel intensities.

The second line contains n space-separated integers representing the pixel intensities.

Output Format

The output displays a tuple containing the absolute differences between consecutive pixel intensities.

Refer to the sample output for format specifications.

Sample Test Case

Input: 5
200 100 20 80 10
Output: (100, 80, 60, 70)

Answer

```
# You are using Python
# Read the number of pixel intensities
n = int(input())

# Read the pixel intensities as a list
pixels = list(map(int, input().split()))

# Calculate the absolute differences between consecutive pixel intensities
differences = tuple(abs(pixels[i] - pixels[i + 1]) for i in range(n - 1))

# Output the result as a tuple
print(differences)
```

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

2. 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 ax2 +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

# Read input values
n = int(input())  # Read the number of coefficients, should always be 3
a, b, c = map(int, input().split())  # Read the coefficients a, b, and c

# Calculate discriminant
D = b**2 - 4*a*c

# Determine the roots based on the discriminant
if D > 0:  # Two distinct real roots
    root1 = (-b + math.sqrt(D)) / (2 * a)
```

```
    root2 = (-b - math.sqrt(D)) / (2 * a)
    print(f"({root1}, {root2})")
elif D == 0:  # One repeated real root
    root = -b / (2 * a)
    print(f"({root})")
else:  # Complex roots
    real_part = -b / (2 * a)
    imaginary_part = math.sqrt(-D) / (2 * a)
    print(f"(({real_part}, {imaginary_part}), ({real_part}, {-imaginary_part}))")
```

Status : Partially correct ***Marks : 7.5/10***

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, w1, and w2, 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
# Read input
w1 = input().strip().lower()
w2 = input().strip().lower()

# Convert words into sets of characters
set_w1 = set(w1)
set_w2 = set(w2)

# Find common letters
common_letters = sorted(set_w1 & set_w2)

# Find unique letters (in w1 but not in w2, and vice versa)
unique_letters = sorted((set_w1 - set_w2) | (set_w2 - set_w1))

# Check if w1 is a superset of w2
is_superset = set_w1 >= set_w2

# Check if there are no common letters
no_common = len(common_letters) == 0
```

```
# Output the results
print(common_letters)
print(unique_letters)
print(is_superset)
print(no_common)
```

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

4. 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
# Read input
n = int(input())  # Size of the first tuple
seq1 = tuple(map(int, input().split()))  # First DNA sequence as a tuple

m = int(input())  # Size of the second tuple
seq2 = tuple(map(int, input().split()))  # Second DNA sequence as a tuple

# Find the minimum length of the two sequences to avoid out of range access
min_length = min(n, m)

# Find the matching bases at the same positions
matching_bases = [seq1[i] for i in range(min_length) if seq1[i] == seq2[i]]

# Output the matching bases as space-separated integers
print(" ".join(map(str, matching_bases)))
```

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