

Student: kamali rj Email id: 240701225@rajalakshmi.edu.in Test: REC_Python_Week 7_CY Course: NeoColab_REC_CS2321_Python Programming

IP Address: 115.245.95.250
Test Duration: 00:12:44

Tab Switches: --
Test Start Time: May 29, 2025 | 03:37 PM

OS Used: Windows
Test Submit Time: May 29, 2025 | 04:51 PM

Browser Used: Firefox
Resume Count: 4

Summary Sections

Filters

1 Coding (5)

Question No: 1

Single File Programming Question

Problem Statement

Rekha is a meteorologist analyzing rainfall data collected over 5 years, with monthly rainfall recorded for each year. She wants to find the total rainfall each year and also identify the month with the maximum rainfall for every year.

Help her to implement the task using the **numpy** package.

Formula:

Yearly total rainfall = sum of all 12 months' rainfall for each year

Month with max rainfall = index of the maximum rainfall value within the 12 months for each year (0-based index)

Input format :

The input consists of 5 lines.

Each line contains 12 floating-point values separated by spaces, representing the rainfall data (in mm) for each month of that year.

Output format :

The first line of output prints: yearly_totals

The second line of output prints: max_rainfall_months

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

Rainfall values are non-negative and can be fractional

Typical rainfall range per month: 0.0 to 1000.0 mm

Sample test cases :

Input 1:

```
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0
2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0
3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0
4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0
```

Output 1:

```
[ 78. 90. 102. 114. 126.]
[11 11 11 11 11]
```

Input 2:

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

Output 2:

```
[0. 0. 0. 0. 0.]
[0 0 0 0 0]
```

Fill your code here

Python ..

```
1 # You are using Python
2 import numpy as np
3
4 # Read the rainfall data for 5 years (each with 12 months)
5 data = [list(map(float, input().split())) for _ in range(5)]
6
7 # Convert the list to a numpy array
8 rainfall = np.array(data)
9
10 # Calculate yearly total rainfall (sum across months for each year)
11 yearly_totals = rainfall.sum(axis=1)
12
13 # Find the month index with maximum rainfall for each year (0-based)
14 max_rainfall_months = rainfall.argmax(axis=1)
15
16 # Print results
17 print(yearly_totals)
18 print(max_rainfall_months)
19
```

Show testcase scores Show solution

Question No: 2

Single File Programming Question

Problem Statement

You are working as a data analyst for a small retail store that wants to track the stock levels of its products. Each product has a unique Name (such as "Toothpaste", "Shampoo", "Soap") and an associated Quantity in stock. Management wants to identify which products have zero stock so they can be restocked.

Write a Python program using the pandas library to help with this task. The program should:

- Read the number of products, n.
- Read n lines, each containing the Name of the product and its Quantity, separated by a space.
- Convert this data into a pandas DataFrame.
- Identify and display the Name and Quantity of products with zero stock.
- If no products have zero stock, display: No products with zero stock.

Input format :

The first line contains an integer n, the number of products.

The next n lines each contain:

<Product_ID> <Quantity>

where <Product_ID> is a single word (e.g., "Shampoo") and <Quantity> is a non-negative integer (e.g., 5).

Output format :

The first line of output prints:

Products with Zero Stock:

If there are any products with zero stock, the following lines print the pandas DataFrame showing those products with two columns: Product_ID and Quantity.

The column headers **Product_ID** and **Quantity** are printed in the second line.

Each subsequent line shows the product's name and quantity, aligned under the respective headers, with no index column.

The output formatting (spacing and alignment) follows the default pandas `to_string(index=False)` style.

If no products have zero stock, print:

No products with zero stock.

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

$1 \leq n \leq 100$

Product names are single English words, no spaces.

Quantities are integers from 0 to 10,000.

Sample test cases :

Input 1:

```
3
P101 10
P102 0
P103 5
```

Output 1:

```
Products with Zero Stock:
Product_ID  Quantity
P102          0
```

Input 2:

```
4
M1 7
M2 8
M3 9
M4 1
```

Output 2:

```
Products with Zero Stock:
No products with zero stock.
```

Fill your code here

Python ...

```
1 # You are using Python
2 import pandas as pd
3
4 # Read number of products
5 n = int(input())
6
7 # Read product data
8 data = [input().split() for _ in range(n)]
9
10 # Create DataFrame
11 df = pd.DataFrame(data, columns=["Product_ID", "Quantity"])
12
13 # Convert Quantity column to integer
14 df["Quantity"] = df["Quantity"].astype(int)
15
16 # Filter products with zero stock
17 zero_stock_df = df[df["Quantity"] == 0]
18
19 # Print output as required
20 print("Products with Zero Stock:")
21 if zero_stock_df.empty:
22     print("No products with zero stock.")
23 else:
```

```
24 print(zero_stock_df.to_string(index=False))
25
```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **2** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Pandas** |
Sub Topic **Pandas Library** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Question No: 3

Single File Programming Question

Problem Statement

Rekha works as an e-commerce data analyst. She receives transaction data containing purchase dates and needs to extract the month and day from these dates using the **pandas** package.

Help her implement this task by performing the following steps:

Convert the Purchase Date column to datetime format, treating invalid date entries as NaT (missing).

Create two new columns:

Purchase Month, containing the month (as an integer) extracted from the Purchase Date.

Purchase Day, containing the day (as an integer) extracted from the Purchase Date. Keep the rest of the data as is.

Input format :

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

The second line contains the CSV header — comma-separated column names.

The next n lines each contain a transaction record in comma-separated format.

Output format :

The first line of output is the text:

Transformed E-commerce Transaction Data:

The next lines print the pandas DataFrame with:

The original columns (including Purchase Date, which is now in datetime format or NaT if invalid).

Two additional columns: Purchase Month and Purchase Day.

The output uses the default pandas DataFrame string representation as produced by print(transformed_df).

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ n ≤ 1000

The CSV data includes at least a Purchase Date column (case-sensitive).

Purchase Date values may include valid and invalid date formats.

Invalid date entries in the Purchase Date column are set to NaT (missing).

The CSV data includes additional columns, which should remain unchanged.

Sample test cases :

Input 1:

```
3
Customer,Purchase Date
Alice,2023-05-15
Bob,2023-06-20
Charlie,2023-07-01
```

Output 1:

Transformed E-commerce Transaction Data:

	Customer	Purchase Date	Purchase Month	Purchase Day
0	Alice	2023-05-15	5	15
1	Bob	2023-06-20	6	20
2	Charlie	2023-07-01	7	1

Input 2:

```
2
Customer,Purchase Date
David,2023-01-10
Emma,2023-12-25
```

Output 2:

Transformed E-commerce Transaction Data:

	Customer	Purchase Date	Purchase Month	Purchase Day
0	David	2023-01-10	1	10
1	Emma	2023-12-25	12	25

Fill your code here

```
1 # You are using Python
2 import pandas as pd
3 from io import StringIO
4
5 # Read number of records
6 n = int(input())
7
8 # Read CSV header
9 header = input()
10
11 # Read the next n lines of CSV data
12 lines = [input() for _ in range(n)]
13
14 # Combine header and data lines into a single CSV string
15 csv_data = "\n".join([header] + lines)
16
17 # Read into pandas DataFrame
18 df = pd.read_csv(StringIO(csv_data))
19
20 # Convert 'Purchase Date' to datetime, invalid parsing will become NaT
21 df['Purchase Date'] = pd.to_datetime(df['Purchase Date'], errors='coerce')
22
23 # Extract month and day from 'Purchase Date'
24 df['Purchase Month'] = df['Purchase Date'].dt.month
25 df['Purchase Day'] = df['Purchase Date'].dt.day
```

Python ...

```

23 df['Purchase Day'] = df['Purchase Date'].dt.day
24
25 # Print output
26 print("Transformed E-commerce Transaction Data:")
27 print(df)
28
29
30

```

Status Partially correct | Mark obtained 6.5/10 | Hints used 0 | Times compiled 2 | Times submitted 2 | Level Medium | Question type Single File Programming | Subject Python |
Topic Pandas | Sub Topic Pandas Library | Blooms taxonomy Apply |

Show testcase scores Show solution

Question No: 4

Single File Programming Question

Problem Statement

Arjun is developing a system to monitor environmental sensors installed in different rooms of a smart building. Each sensor records multiple temperature readings throughout the day. To compare sensor data fairly despite differing scales, Arjun needs to normalize each sensor's readings so that they have a mean of zero and standard deviation of one.

Help him implement this normalization using **numpy**.

Normalization Formula:

$$\text{Normalized value} = \frac{\text{value} - \text{mean of sensor readings}}{\text{standard deviation of sensor readings}}$$

Input format :

The first line of input consists of two integers: sensors (number of sensors) and samples (number of readings per sensor).

The next sensors lines each contain samples space-separated floats representing the sensor readings.

Output format :

The first line of output prints: "Normalized Sensor Data:"

The next lines print the normalized readings as a numpy array, where each row corresponds to a sensor's normalized values.

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ sensors, samples ≤ 1000

Sensor readings are real numbers.

Sample test cases :

Input1:

```

3 3
1.0 2.0 3.0
4.0 5.0 6.0
7.0 8.0 9.0

```

Output1:

```

Normalized Sensor Data:
[[ -1.22474487  0.           1.22474487]
 [ -1.22474487  0.           1.22474487]
 [ -1.22474487  0.           1.22474487]]

```

Input 2:

```

2 4
1.0 2.0 3.0 4.0
4.0 3.0 2.0 1.0

```

Output 2:

```

Normalized Sensor Data:
[[ -1.34164079 -0.4472136   0.4472136   1.34164079]
 [  1.34164079  0.4472136  -0.4472136  -1.34164079]]

```

Whitelist

Set I: **numpy**

Fill your code here

```

1 # You are using Python
2 import numpy as np
3
4 # Read input
5 sensors, samples = map(int, input().split())
6 data = [list(map(float, input().split())) for _ in range(sensors)]
7
8
9 arr = np.array(data)
10
11 # Compute mean and std for each sensor (row-wise)
12 mean = arr.mean(axis=1, keepdims=True)
13 std = arr.std(axis=1, keepdims=True)
14
15 # Normalize: (value - mean) / std
16 normalized = (arr - mean) / std
17
18 print("Normalized Sensor Data:", normalized)
19

```

Python ...

Show testcase scores Show solution

Question No: 5

Single File Programming Question

Problem Statement

Arjun is monitoring hourly temperature data recorded continuously for multiple days. He needs to calculate the average temperature for each day based on 24 hourly readings.

Help him to implement the task using the **numpy** package.

Formula:

Reshape the temperature readings into rows where each row has 24 readings (one day).

Average temperature per day = mean of 24 hourly readings in each row.

Input format :

The first line of input consists of an integer value, n, representing the total number of temperature readings.

The second line of input consists of n floating-point values separated by spaces, representing hourly temperature readings.

Output format:

The output prints: avg_per_day

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

n is a multiple of 24

24 ≤ n ≤ 2400

Sample test cases :

Input 1:

```
30  
30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0
```

Output 1:

```
[30.]
```

Input 2:

```
24  
-5.0 -4.0 -3.0 -2.0 -1.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 -5.0 -4.0 -3.0 -2.0 -1.0 0.0 1.0 2.0 3.0
```

Output 2:

```
[0.5]
```

Input 3:

```
72  
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.
```

Output 3:

```
[12.5 36.5 60.5]
```

Fill your code here

```
1 # You are using Python  
2 import numpy as np  
3  
4 # Read the total number of temperature readings  
5 n = int(input())  
6  
7 # Read the temperature readings  
8 temps = list(map(float, input().split()))  
9  
10 # Convert to numpy array  
11 arr = np.array(temps)  
12  
13 # Reshape into days (each with 24 readings)  
14 days = arr.reshape(-1, 24)  
15  
16 # Calculate average temperature per day  
17 avg_per_day = days.mean(axis=1)  
18  
19 # Print the result  
20 print(avg_per_day)  
21
```

Python ... 

Student: kamali rj Email id: 240701225@rajalakshmi.edu.in Test: REC_Python_Week 7_PAH Course: NeoColab_REC_CS23221_Python Programming

IP Address: 115.245.95.250
Test Duration: 00:43:44Tab Switches: --
Test Start Time: May 29, 2025 | 04:01 PMOS Used: Windows
Test Submit Time: May 29, 2025 | 04:50 PMBrowser Used: Firefox
Resume Count: 4

Summary Sections

Filters

Coding (5)

Question No: 1

Single File Programming QuestionProblem Statement

A software development company wants to classify its employees based on their years of service at the company. They want to categorize employees into three experience levels: Junior (less than 3 years), Mid (3 to 6 years, inclusive), and Senior (more than 6 years).

Experience Level Classification:

Junior: Years at Company < 3
Mid: 3 ≤ Years at Company < 6
Senior: Years at Company > 6

You need to create a Python program using the **pandas** library that reads employee data, processes it into a DataFrame, and adds a new column "Experience Level" to display the appropriate classification for each employee.

Input format:

First line: an integer n representing the number of employees.

Next n lines: each line has a string Name and a floating-point number Years at Company (space-separated).

Output format:

First line: "Employee Data with Experience Level:"

The employee data table printed with no index column, and with columns: Name, Years at Company, Experience Level.

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ n ≤ 100 – Number of employees will not exceed 100.

0 ≤ Years at Company ≤ 50 – Years of service will always be a non-negative number (up to 50).

Names are single-word strings containing only English alphabets (no spaces).

Input data is well-formed, so you do not need to handle malformed inputs.

Sample test cases :**Input1:**

```
5
Alice 2
Bob 4
Charlie 7
Diana 3
Evan 6
```

Output1:

Employee Data with Experience Level:		
Name	Years at Company	Experience Level
Alice	2.0	Junior
Bob	4.0	Mid
Charlie	7.0	Senior
Diana	3.0	Mid
Evan	6.0	Senior

Input2:

```
3
John 1.5
Jane 5
Mark 10
```

Output2:

Employee Data with Experience Level:		
Name	Years at Company	Experience Level
John	1.5	Junior
Jane	5.0	Mid
Mark	10.0	Senior

Fill your code here

```
1 # You are using Python
2 import pandas as pd
3 import sys
4
5 n = int(sys.stdin.readline().strip())
6 data = []
7
8 for _ in range(n):
9     line = sys.stdin.readline().strip().split()
10    name = line[0]
11    years = float(line[1])
12    data.append([name, years])
13
14 df = pd.DataFrame(data, columns=["Name", "Years at Company"])
15
16 def classify(years):
17    if years < 3:
18        return "Junior"
19    elif 3 <= years < 6:
20        return "Mid"
21    else:
22        return "Senior"
23
24 df["Experience Level"] = df["Years at Company"].apply(classify)
25 print("Employee Data with Experience Level:")
```

Python ...

```
25 print('Employee Data with Experience Level: ')
26 print(df.to_string(index=False))
27
28
```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **5** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Pandas** |
Sub Topic **Pandas Library** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Question No: 2

Single File Programming Question

Problem Statement

Arjun manages a busy customer service center and wants to analyze the distribution of customer wait times to improve service efficiency. He decides to group the wait times into intervals of 5 minutes each and count how many customers fall into each interval bucket.

Help him implement this bucketing and counting task using **NumPy**.

Bucketing Logic:

Divide the wait times into intervals (buckets) of size 5 minutes, e.g.:
[0-5), [5-10), [10-15), ...

Use NumPy's digitize function to determine which bucket each wait time falls into.
Count the number of wait times in each bucket and generate bucket labels.

Input format :

The first line contains an integer n, the number of customer wait times recorded.

The second line contains n space-separated floating-point numbers representing the wait times (in minutes).

Output format :

The first line of output is the text:

Wait Time Buckets and Counts:

Each subsequent line prints the bucket range and the number of wait times in that bucket, formatted as:

<bucket_range>: <count>
where <bucket_range> is the lower and upper bound of the bucket (inclusive lower bound, exclusive upper bound), for example:

0-5: 3

5-10: 2

10-15: 1

The output uses the default string formatting of Python's print() function (no extra spaces, no special formatting beyond the specified lines).

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ n ≤ 100

Wait times are non-negative floating-point numbers.

Bucket size is fixed at 5 minutes.

Buckets are generated dynamically up to the largest wait time (rounded up to the nearest bucket).

Sample test cases :

Input1:

```
10
2.0 3.0 7.0 8.0 12.0 14.0 18.0 19.0 21.0 25.0
```

Output1:

```
Wait Time Buckets and Counts:
0-5: 2
5-10: 2
10-15: 2
15-20: 2
20-25: 1
```

Input2:

```
5
1.0 6.0 11.0 16.0 21.0
```

Output2:

```
Wait Time Buckets and Counts:
0-5: 1
5-10: 1
10-15: 1
15-20: 1
20-25: 1
```

Fill your code here

```
1
2 import numpy as np
3 import math
4
5
6 n = int(input())
7 wait_times = np.array(list(map(float, input().split())))
8
9
10 bucket_size = 5
11 max_time = wait_times.max()
12 num_buckets = math.ceil(max_time / bucket_size)
13
14 bins = np.arange(0, (num_buckets + 1) * bucket_size, bucket_size)
15
16
17 bucket_indices = np.digitize(wait_times, bins, right=False)
```

Python .. 

```

19 counts = np.zeros(num_buckets, dtype=int)
20 for idx in bucket_indices:
21
22     if 1 <= idx < num_buckets:
23         counts[idx - 1] += 1
24
25
26 print("Wait Time Buckets and Counts:")
27 for i in range(num_buckets):
28     lower = bins[i]
29     upper = bins[i + 1]
30     print(f"{int(lower)}-{int(upper)}: {counts[i]}")
31

```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **12** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Numpy** | Sub Topic **Numpy Library in python** | Blooms taxonomy **Apply** | [Apply](#)

Show testcase scores Show solution

Question No: 3

Single File Programming Question

Problem Statement

A company conducted a customer satisfaction survey where each respondent provides their RespondentID and an optional textual Feedback. Sometimes, respondents submit their ID without any feedback or with empty feedback.

Your task is to process the survey responses using **pandas** to replace any missing or empty feedback with the phrase "No Response". Finally, print the cleaned survey responses exactly as shown in the sample output.

Input format :

The first line contains an integer n, the number of survey responses.

Each of the next n lines contains:

A RespondentID (a single alphanumeric string without spaces),

Followed optionally by a Feedback string, which may be empty or missing.

If no feedback is provided after the RespondentID, treat it as missing.

Output format :

Print the line:

Survey Responses with Missing Feedback Filled:

Then print the cleaned survey data as a table with two columns: RespondentID and Feedback.

The table should have the headers exactly as:

RespondentID Feedback

Print each respondent's data on a new line, aligned to match the output produced by `pandas.DataFrame.to_string(index=False)`.

For any missing or empty feedback, print "**No Response**" in the Feedback column.

Maintain the spacing and alignment exactly as shown in the sample outputs.

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

$1 \leq n \leq 1000$

RespondentID consists of alphanumeric characters without spaces.

Feedback may be any string, possibly containing spaces, or may be missing entirely.

Sample test cases :

Input 1:

```

4
101 Great service
102
103 Loved it
104

```

Output 1:

```

Survey Responses with Missing Feedback Filled:
RespondentID      Feedback
101      Great service
102      No Response
103      Loved it
104      No Response

```

Input 2:

```

3
201
202 Okay experience
203

```

Output 2:

```

Survey Responses with Missing Feedback Filled:
RespondentID      Feedback
201      No Response
202      Okay experience
203      No Response

```

Fill your code here

```

1 # You are using Python
2 import pandas as pd
3 import sys
4
5 n = int(sys.stdin.readline().strip())
6
7 data = []
8 for _ in range(n):
9     line = sys.stdin.readline().rstrip('\n')
10    # Split only on first space to separate RespondentID and Feedback (if any)
11    parts = line.split(' ', 1)
12    respondent_id = parts[0]
13    feedback = parts[1].strip() if len(parts) > 1 else ''
14    data.append([respondent_id, feedback])

```

Python .. 

```

15
16 df = pd.DataFrame(data, columns=['RespondentID', 'Feedback'])
17
18 # Replace empty strings and missing with "No Response"
19 df['Feedback'] = df['Feedback'].replace('', 'No Response')
20
21 print("Survey Responses with Missing Feedback Filled:")
22 print(df.to_string(index=False))
23

```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **1** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Pandas** |
Sub Topic **Pandas Library** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Question No: 4

Single File Programming Question

Problem Statement

Arjun is a data scientist working on an image processing task. He needs to normalize the pixel values of a grayscale image matrix to scale between 0 and 1. The input image data is provided as a matrix of integers.

Help him to implement the task using the **numpy** package.

Formula:

To normalize each pixel value in the image matrix:

normalized_pixel = (pixel - min_pixel) / (max_pixel - min_pixel)
where min_pixel and max_pixel are the minimum and maximum pixel values in the image matrix, respectively. If all pixel values are the same, the normalized image matrix should be filled with zeros.

Input format:

The first line of input consists of an integer value, rows, representing the number of rows in the image matrix.

The second line of input consists of an integer value, cols, representing the number of columns in the image matrix.

The next rows lines each consist of cols integer values separated by a space, representing the pixel values of the image matrix.

Output format:

The output prints: normalized_image

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ rows ≤ 100

1 ≤ cols ≤ 100

0 ≤ pixel ≤ 255

Sample test cases :

Input 1:

```

2
3
1 2 3
4 5 6

```

Output 1:

```

[[0.  0.2 0.4]
 [0.6 0.8 1. ]]

```

Input 2:

```

2
2
0 0
0 0

```

Output 2:

```

[[0. 0.]
 [0. 0.]]

```

Fill your code here

Python ...

```

1 import numpy as np
2
3
4 rows = int(input())
5 cols = int(input())
6
7
8 matrix = []
9 for _ in range(rows):
10     row = list(map(int, input().split()))
11     matrix.append(row)
12
13
14 image = np.array(matrix, dtype=float)
15
16
17 min_pixel = image.min()
18 max_pixel = image.max()
19
20
21 if min_pixel == max_pixel:
22     normalized_image = np.zeros_like(image)
23 else:
24     normalized_image = (image - min_pixel) / (max_pixel - min_pixel)
25

```

```
26  
27  
28 print(normalized_image)  
29
```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **4** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Numpy** |
Sub Topic **Numpy Library in python** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Question No: 5

Single File Programming Question

Problem Statement

You're analyzing the daily returns of a set of financial assets over a period of time. Each day is represented as a row in a 2D array, where each column represents the return of a specific asset on that day.

Your task is to identify which days had all positive returns across every asset using **numpy**, and output a boolean array indicating these days.

Input format:

The first line of input consists of two integer values, rows and cols, separated by a space.

Each of the next rows lines consists of cols float values representing the returns of the assets for that day.

Output format:

The first line of output prints: "**Days where all asset returns were positive:**"

The second line of output prints: the boolean array **positive_days**, indicating True for days where all asset returns were positive and False otherwise.

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ rows ≤ 1000

1 ≤ cols ≤ 1000

Each asset return is a float number (can be positive, negative, or zero).

Sample test cases :

Input1:

```
3 4  
0.01 0.02 0.03 0.04  
0.05 0.06 0.07 0.08  
-0.01 0.02 0.03 0.04
```

Output1:

```
Days where all asset returns were positive:  
[ True  True False]
```

Input2:

```
2 3  
0.1 -0.2 0.3  
0.0 0.1 -0.2
```

Output2:

```
Days where all asset returns were positive:  
[False False]
```

Fill your code here

Python ..  

```
1 # You are using Python  
2 import numpy as np  
3  
4 # Read dimensions  
5 rows, cols = map(int, input().split())  
6  
7 # Read data for each day  
8 data = []  
9 for _ in range(rows):  
10     row = list(map(float, input().split()))  
11     data.append(row)  
12  
13 # Convert to NumPy array  
14 returns = np.array(data)  
15  
16 # Identify days where all returns are positive  
17 positive_days = np.all(returns > 0, axis=1)  
18  
19 # Print result  
20 print("Days where all asset returns were positive:")  
21 print(positive_days)  
22
```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **1** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Numpy** |
Sub Topic **Numpy Library in python** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Student: kamali rj Email id: 240701225@rajalakshmi.edu.in Test: REC_Python_Week 7_COD Course: NeoColab_REC_CS2321_Python Programming

IP Address: 115.245.95.250

Test Duration: 00:09:07

Tab Switches: --

Test Start Time: May 29, 2025 | 04:30 PM

OS Used: Windows

Test Submit Time: May 29, 2025 | 04:57 PM

Browser Used: Firefox

Resume Count: 3

Summary Sections

Filters

1 Coding (5)

Question No: 1

Single File Programming Question

Problem Statement

A company tracks the monthly sales data of various products. You are given a table where each row represents a product and each column represents its monthly sales in sequential months.

Your task is to compute the cumulative monthly sales for each product using **numpy**, where the cumulative sales for a month is the total sales from month 1 up to that month.

Input format:

The first line of input consists of two integer values, products and months, separated by a space.

Each of the next products lines consists of months integer values representing the monthly sales data of a product.

Output format:

The first line of output prints: "Cumulative Monthly Sales:"

The second line of output prints: the 2D numpy array cumulative_array that contains the cumulative sales data for each product.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

1 ≤ products ≤ 1000

1 ≤ months ≤ 1000

Each sales value is a non-negative integer (0 ≤ sales ≤ 10000)

Sample test cases:

Input 1:

```
2 4
10 20 30 40
5 15 25 35
```

Output 1:

```
Cumulative Monthly Sales:
[[ 10 30 60 100]
 [ 5 20 45 80]]
```

Input 2:

```
3 3
1 2 3
4 5 6
7 8 9
```

Output 2:

```
Cumulative Monthly Sales:
[[ 1 3 6]
 [ 4 9 15]
 [ 7 15 24]]
```

Fill your code here

Python ..

```
1 # You are using Python
2 import numpy as np
3
4 # Read number of products and months
5 products, months = map(int, input().split())
6
7 # Read monthly sales data for each product
8 data = [list(map(int, input().split())) for _ in range(products)]
9
10 # Convert to numpy array
11 sales_array = np.array(data)
12
13 # Compute cumulative sum along months (axis=1)
14 cumulative_array = np.cumsum(sales_array, axis=1)
15
16 # Print output
17 print("Cumulative Monthly Sales:", cumulative_array)
18
```

Question No: 2

Single File Programming Question

Problem Statement

Sita is analyzing her company's daily sales data to find all sales values that are multiples of 5 and exceed 100. She wants to filter these specific sales values from the list.

Help her to implement the task using the **numpy** package.

Formula:

To filter sales values:

Select all values s from sales such that $(s \% 5 == 0)$ and $(s > 100)$

Input format :

The first line of input consists of an integer value, n , representing the number of sales entries.

The second line of input consists of n floating-point values, sales, separated by spaces, representing daily sales figures.

Output format :

The output prints: filtered_sales

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

$1 \leq n \leq 15$

$0 \leq \text{sales}[i] \leq 1000$

Sample test cases :

Input 1:

```
5
50.0 100.0 105.0 150.0 99.0
```

Output 1:

```
[105. 150.]
```

Input 2:

```
6
200.0 205.0 99.0 120.0 125.0 88.0
```

Output 2:

```
[200. 205. 120. 125.]
```

Fill your code here

Python .. ▾

```
1 # You are using Python
2 import numpy as np
3
4 # Read number of sales entries
5 n = int(input())
6
7 # Read sales data as floats
8 sales = np.array(list(map(float, input().split())))
9
10 # Filter sales where (value % 5 == 0) and (value > 100)
11 filtered_sales = sales[(sales % 5 == 0) & (sales > 100)]
12
13 # Print filtered sales
14 print(filtered_sales)
15
```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **1** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Numpy** |
Sub Topic **Numpy Library in python** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Question No: 3

Single File Programming Question

Problem Statement

Alex is a data scientist analyzing the relationship between two financial indicators over time. He has collected two time series datasets representing daily values of these indicators over several months. Alex wants to understand how these two indicators correlate at different time lags to identify possible leading or lagging behaviors.

Your task is to help Alex compute the cross-correlation of these two time series using **numpy**, so he can analyze the similarity between the two signals at various time shifts.

Input format :

The first line of input consists of space-separated float values representing the first time series, array1.

The second line of input consists of space-separated float values representing the second time series, array2.

Output format:

The first line of output prints: "Cross-correlation of the two time series:"

The second line of output prints: the 1D numpy array cross_corr representing the cross-correlation of array1 and array2 across different lags.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

$1 \leq \text{len(array1)} == \text{len(array2)} \leq 1000$

The input values in the arrays are floating-point numbers (can be positive, negative, or zero).

Sample test cases:

Input 1:

```
1.0 2.0 3.0  
4.0 5.0 6.0
```

Input 2:

```
-1.0 0.0 1.0  
1.0 0.0 -1.0
```

Output 1:

```
Cross-correlation of the two time series:  
[ 6. 17. 32. 23. 12.]
```

Output 2:

```
Cross-correlation of the two time series:  
[ 1. 0. -2. 0. 1.]
```

Fill your code here

```
1 # You are using Python  
2 import numpy as np  
3  
4 # Read input  
5 array1 = np.array(list(map(float, input().split())))  
6 array2 = np.array(list(map(float, input().split())))  
7  
8 # Compute cross-correlation  
9 cross_corr = np.correlate(array1, array2, mode='full')  
10  
11 # Output  
12 print("Cross-correlation of the two time series:")  
13 print(cross_corr)  
14
```

Python ...

Status	Correct	Mark obtained	10/10	Hints used	0	Times compiled	4	Times submitted	1	Level	Medium	Question type	Single File Programming	Subject	Python	Topic	Pandas
Sub Topic	Pandas Library	Blooms taxonomy	Apply														

Show testcase scores Show solution

Question No: 4

Single File Programming Question

Problem Statement

Sita works as a sales analyst and needs to analyze monthly sales data for different cities. She receives lists of cities, months, and corresponding sales values and wants to create a **pandas** DataFrame using a MultiIndex of cities and months.

Help her to implement this task and calculate total sales for each city.

Input format:

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

The second line of input consists of n space-separated city names.

The third line of input consists of n space-separated month names.

The fourth line of input consists of n space-separated float values representing sales for each city-month combination.

Output format:

The first line of output prints: "Monthly Sales Data with MultiIndex:"

The next lines print the DataFrame with MultiIndex (City, Month) and their corresponding sales values.

The following line prints: "\nTotal Sales Per City:"

The final lines print the total sales per city, computed by grouping the sales data on city names.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

$1 \leq n \leq 1000$

Cities and months are provided as strings.

Sales values are floats or integers.
Each city-month combination is unique.

Sample test cases :

Input 1:

```
4  
NYC NYC LA LA  
Jan Feb Jan Feb  
100 200 300 400
```

Output 1:

```
Monthly Sales Data with MultiIndex:  
Sales  
City Month  
NYC Jan 100.0  
Feb 200.0  
LA Jan 300.0  
Feb 400.0
```

Total Sales Per City:

```
Sales  
City  
LA 700.0  
NYC 300.0
```

Input 2:

```
6  
Boston Boston Miami Miami Miami Boston  
Mar Apr May May Mar Apr  
50 60 70 80 90 100
```

Output 2:

```
Monthly Sales Data with MultiIndex:  
Sales  
City Month  
Boston Mar 50.0  
Apr 60.0  
Miami May 70.0  
May 80.0  
Mar 90.0  
Boston Apr 100.0
```

Total Sales Per City:

```
Sales  
City  
Boston 210.0  
Miami 240.0
```

Fill your code here

Python .. ▾

```
1 # You are using Python  
2 import pandas as pd  
3  
4 # Read input  
5 n = int(input())  
6 cities = input().split()  
7 months = input().split()  
8 sales = list(map(float, input().split()))  
9  
10 # Create a MultiIndex from cities and months  
11 index = pd.MultiIndex.from_tuples(list(zip(cities, months)), names=["City", "Month"])  
12  
13 # Create the DataFrame  
14 df = pd.DataFrame(sales, index=index, columns=["Sales"])  
15  
16 # Print DataFrame with MultiIndex  
17 print("Monthly Sales Data with MultiIndex:")  
18 print(df)  
19  
20 # Calculate total sales per city by grouping on 'City'  
21 total_sales = df.groupby(level="City").sum()  
22  
23 print("\nTotal Sales Per City:")  
24 print(total_sales)  
25
```

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **1** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Pandas** |
Sub Topic **Pandas Library** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Question No: 5

Single File Programming Question

Problem Statement

Rekha works in hospital data management and receives patient records with missing or incomplete data. She needs to clean the records by performing the following tasks:

- Calculate the mean of the available Age values.
- Replace any missing (NaN) values in the Age column with this mean age.
- Remove any rows where the Diagnosis value is missing (NaN).
- Reset the DataFrame index after removing these rows.

Implement this data cleaning task using the **pandas** package.

Input format :

The first line of input contains an integer n representing the number of patient records.

The second line contains the CSV header – comma-separated column names (e.g., "Name,Age,Diagnosis,Gender").

The next n lines each contain one patient record in comma-separated format.

Output format:

The first line of output is the text:

Cleaned Hospital Records:

The next lines print the cleaned pandas DataFrame (as produced by `print(cleaned_df)`).

This will include the updated values of the Age column (with missing ages filled by the mean age), and any rows with missing Diagnosis removed.

The DataFrame will be displayed using the default pandas `print()` representation.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

$1 \leq n \leq 1000$

The columns include at least Age and Diagnosis among others.

Age values can be integers or floats, with some missing (NaN).

Diagnosis values can be strings, some may be missing (NaN).

Sample test cases:**Input 1:**

```
5
PatientID,Name,Age,Diagnosis
1,John Doe,45,Flu
2,Jane Smith,,Cold
3,Bob Lee,50,
4,Alice Green,38,Fever
5,Tom Brown,,Infection
```

Output 1:

PatientID	Name	Age	Diagnosis
0	John Doe	45.00000	Flu
1	Jane Smith	44.33333	Cold
2	Alice Green	38.00000	Fever
3	Tom Brown	44.33333	Infection

Input 2:

```
5
PatientID,Name,Age,Diagnosis
101,Mike Ross,29,Asthma
102,Rachel Zane,,
103,Harvey Specter,47,Cancer
104,Donna Paulsen,42,
105,Louis Litt,36,Allergy
```

Output 2:

PatientID	Name	Age	Diagnosis
0	Mike Ross	29.0	Asthma
1	Harvey Specter	47.0	Cancer
2	Louis Litt	36.0	Allergy

Fill your code here

```
1 # You are using Python
2 import pandas as pd
3 import sys
4 import numpy as np
5
6 # Read number of records
7 n = int(input())
8
9 # Read CSV header line and column names
10 header = input().strip()
11
12 # Read all records lines into a list
13 data_lines = [input().strip() for _ in range(n)]
14
15 # Combine header and data lines into a CSV string
16 csv_data = '\n'.join([header] + data_lines)
17
18 # Read into pandas DataFrame using read_csv from a string buffer
19 from io import StringIO
20 df = pd.read_csv(StringIO(csv_data), na_values=[''])
21
22 # Calculate mean of Age column ignoring NaNs
23 mean_age = df['Age'].mean()
24
25 # Replace NaN ages with the mean age
26 df['Age'].fillna(mean_age, inplace=True)
27
28 # Remove rows where Diagnosis is missing (NaN or empty after strip)
29 # Note: We consider missing if Diagnosis is NaN or empty string after stripping whitespace
30 df['Diagnosis'] = df['Diagnosis'].replace('^\\s*$', np.nan, regex=True)
31 df.dropna(subset=['Diagnosis'], inplace=True)
32
33 # Reset index after dropping rows
```

Python .. ▾

Status **Correct** | Mark obtained **10/10** | Hints used **0** | Times compiled **3** | Times submitted **1** | Level **Medium** | Question type **Single File Programming** | Subject **Python** | Topic **Pandas** |

Sub Topic **Pandas Library** | Blooms taxonomy **Apply** |

Show testcase scores Show solution

Student: kamali rj Email id: 240701225@rajalakshmi.edu.in Test: REC_Python_Week 7_Mcq Course: NeoColab_REC_CS2321_Python Programming

IP Address: 2409:40f4:41d:fab9:1f9:26e3:e0b5:5404

Test Duration: 00:23:32

Tab Switches: --

Test Start Time: May 29, 2025 | 04:20 PM

OS Used: Windows

Test Submit Time: May 31, 2025 | 11:35 PM

Browser Used: Firefox

Resume Count: 1

Summary Sections

Filters

1 MCQ (20)

Question No: 11

Multi Choice Type Question

Which NumPy function is used to calculate the standard deviation of an array?

- numpy.average()
- numpy.mean()
- numpy.std()
- numpy.var()

Status Correct | Mark obtained 1/1 | Hints used 0 | Level Easy | Question type MCQ Single Correct | Subject Python | Topic Library | Sub Topic NumPy | Blooms taxonomy Understand |

Show solution

Question No: 12

Multi Choice Type Question

What is the primary purpose of Pandas DataFrame?

- To perform arithmetic operations on data
- None of the mentioned options
- To store data in tabular form for analysis and manipulation
- To create visualizations of data

Status Correct | Mark obtained 1/1 | Hints used 0 | Level Easy | Question type MCQ Single Correct | Subject Python | Topic Pandas | Sub Topic Pandas Dataframe |
Blooms taxonomy Understand |

Show solution

Question No: 13

Multi Choice Type Question

What does the np.arange(10) function in NumPy do?

- Creates an array with values from 0 to 10
- Creates an array with 10 elements, all initialized to 0
- Creates an array with 10 elements, from 1 to 10
- Creates an array with values from 1 to 9

Status Wrong | Mark obtained 0/1 | Hints used 0 | Level Medium | Question type MCQ Single Correct | Subject Python | Topic Library | Sub Topic NumPy | Blooms taxonomy Understand |

Question No: 14

Multi Choice Type Question

Which NumPy function is used to find the indices of the maximum and minimum values in an array?

argmax() and argmin()

index_max() and index_min()

max_index() and min_index()

maximum() and minimum()

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Question No:15

Multi Choice Type Question

In NumPy, how do you access the first element of a one-dimensional array arr?

arr[1]

arr.first()

arr[0]

arr(0)

Status **Wrong** | Mark obtained **0/1** | Hints used **0** | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Show solution

Question No:16

Multi Choice Type Question

What will be the output of the following code snippet?

```
1 import numpy as np
2 arr = np.array([1, 2, 3])
3 result = np.concatenate((arr, arr))
4 print(result)
```

[121121]

[123123]

[321123]

[321321]

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Medium** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Question No:17

Multi Choice Type Question

What is the output of the following code?

```
1 import numpy as np
2 a = np.arange(10)
3 print(a[2:5])
```

[2,3,4]

[0,1,2]

[5,6,7]

[2,4,6]

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Medium** | Question type **MCQ Single Correct** | Subject **Python** | Topic **NumPy** | Sub Topic **NumPy array operations** |

Show solution

Question No:18

Multi Choice Type Question

What will be the output of the following code?

```
1 import pandas as pd
2 pd.Series([1,2], index= ['a','b','c'])
```

- Value Error ✓
- Syntax Error
- None of the mentioned options
- Index Error

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Pandas** | Sub Topic **Pandas Dataframe** | Blooms taxonomy **Understand**

Question No:19

Multi Choice Type Question

What is the output of the following NumPy code snippet?

```
1 import numpy as np
2 arr = np.array([1, 2, 3, 4, 5])
3 r = arr[arr > 2]
4 print(r)
```

- [3 4 5] ✓
- [1 2]
- [1]
- [2 3 4 5]

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Medium** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand**

Question No: 20

Multi Choice Type Question

The important data structure of pandas is/are _____.

- Both Series and Data Frame ✓
- None of the mentioned options
- Data Frame
- Series

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Pandas** | Sub Topic **Pandas Dataframe** | Blooms taxonomy **Understand**

 Show solution

Student: kamali rj Email id: 240701225@rajalakshmi.edu.in Test: REC_Python_Week 7_Mcq Course: NeoColab_REC_CS2321_Python Programming

IP Address: 2409:40f4:411d:fab9:1f9:26e3:e0b5:5404

Test Duration: 00:23:32

Tab Switches: --

Test Start Time: May 29, 2025 | 04:20 PM

OS Used: Windows

Test Submit Time: May 31, 2025 | 11:35 PM

Browser Used: Firefox

Resume Count: 1

Summary Sections

Filters

1 MCQ (20)

Question No: 1

Multi Choice Type Question

Which function is used to create a Pandas DataFrame?

pd.DataFrame()



pd.List()

pd.Series()

pd.array()

Status **Correct** | Mark obtained 1/1 | Hints used 0 | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Pandas** | Sub Topic **Pandas Dataframe**

Blooms taxonomy **Understand**

Show solution

Question No: 2

Multi Choice Type Question

In the DataFrame created in the code, what is the index for the row containing the data for 'Jack'?

```
1 import pandas as pd
2
3 data = {'Name': ['Tom', 'Jack', 'nick', 'juli'],
4         'marks': [99, 98, 95, 90]}
5
6 df = pd.DataFrame(data, index=['rank1',
7                      'rank2',
8                      'rank3',
9                      'rank4'])
10 print(df)
```

rank3

rank4

rank1

rank2



Status **Correct** | Mark obtained 1/1 | Hints used 0 | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Pandas** | Sub Topic **Pandas Dataframe**

Blooms taxonomy **Understand**

Show solution

Question No: 3

Multi Choice Type Question

Minimum number of argument we require to pass in pandas series ?

0

3

1



Question No: 4

Multi Choice Type Question

What does NumPy stand for?

- Nonlinear Python
- Natural Python
- Numerical Python
- Numeric Program

Show solution

Question No: 5

Multi Choice Type Question

What is the output of the following NumPy code?

```
1 import numpy as np
2 arr = np.array([1, 2, 3, 4, 5])
3 r = arr[2:4]
4 print(r)
```

- [2 3]
- [2 3 4]
- [1 2 3]
- [3 4]

Show solution

Question No: 6

Multi Choice Type Question

Which of the following is a valid way to import NumPy in Python?

- import numpy as np
- import np
- import numpy
- from numpy import *

Question No: 7

Multi Choice Type Question

What is the primary data structure used in NumPy for numerical computations?

- Tuple

Array

List

Dictionary

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Show solution

Question No: 8

Multi Choice Type Question

What is the result of the following NumPy operation?

```
1 import numpy as np
2 arr = np.array([1, 2, 3])
3 r = arr + 5
4 print(r)
```

[1 2 3]

Compile Time Error

[6 7 8]

[1 2 3 5]

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Medium** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Show solution

Question No: 9

Multi Choice Type Question

What is the purpose of the following NumPy code snippet?

```
1 import numpy as np
2 arr = np.zeros((3, 4))
3 print(arr)
```

Displays a 3x4 matrix filled with zeros

Displays a 3x4 matrix filled with random values

Displays an empty array

Displays a 3x4 matrix filled with ones

Status **Correct** | Mark obtained **1/1** | Hints used **0** | Level **Medium** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Question No: 10

Multi Choice Type Question

Which NumPy function is used to create an identity matrix?

numpy.ones()

numpy.eye()

numpy.identity()

numpy.zeros()

Status **Wrong** | Mark obtained **0/1** | Hints used **0** | Level **Easy** | Question type **MCQ Single Correct** | Subject **Python** | Topic **Library** | Sub Topic **NumPy** | Blooms taxonomy **Understand** |

Student: kamalraj - Email: 240701225@rejulakshmi.edu.in - Test: RDC_Python_Week 5_CF - Course: NeoColab_RDC_C923231_Python Programming

IP Address: 10.246.82.212 Last login: May 23 10:21:18 from 10.246.82.212
 Last session duration: 00:00:18

Total switches: 1
 Last start time: May 23 10:21:08.18

OS load: 0.00 0.00 0.00
 Last alarm time: May 23 10:21:08.18 (00:00 PM)

Driver load: 0.00 0.00 0.00
 Max queue count: 1

summary Section

Edit Reset

Coding (0) ▾

Question No. 1

single File Programming Question**Problem Statement:**

Implement a program that checks whether a set of three input values can form the sides of a valid triangle. The program defines a function `is_valid_triangle` that takes three side lengths as arguments and raises a `ValueError` if any side length is not a positive value. It then checks whether the sum of any two sides is greater than the third side to determine the validity of the triangle.

Input Format:

The first line of input consists of an integer `A`, representing side1.
 The second line of input consists of an integer `B`, representing side2.
 The third line of input consists of an integer `C`, representing side3.

Output Format:

The output prints either "It's a valid triangle" if the input side lengths form a valid triangle,
 or "It's not a valid triangle" if they do not.
 If there is a `ValueError`, it should print "ValueError: error_message".

Refer to the sample output for the formatting specifications.

Code constraints:

In this scenario, the given test cases will fail under the following conditions:

`1 <= side1, side2, side3 <= 1000`

Sample test cases:**Input 1:**

```
3
4
5
```

Output 1:

```
It's a valid triangle
```

Input 2:

```
1
3
5
```

Output 2:

```
It's not a valid triangle
```

Input 3:

```
4
-2
5
```

Output 3:

```
ValueError: Side lengths must be positive
```

Input 4:

```
7
5
9
```

Output 4:

```
ValueError: Side lengths must be positive
```

All your code here:

```
#! /usr/bin/python3
# Check if the given sides form a valid triangle
# If any side is not a positive number, raise ValueError("Side lengths must be positive")
# If A + B > C and A + C > B and B + C > A, return "It's a valid triangle"
# Else, return "It's not a valid triangle"

def is_valid_triangle(a, b, c):
    """Check if the given sides form a valid triangle"""
    if a < 0 or b < 0 or c < 0:
        raise ValueError("Side lengths must be positive")
    if a + b > c and a + c > b and b + c > a:
        return "It's a valid triangle"
    else:
        return "It's not a valid triangle"

if __name__ == "__main__":
    a = int(input())
    b = int(input())
    c = int(input())
    print(is_valid_triangle(a, b, c))

except ValueError as e:
    print("ValueError: " + str(e))
```

Status	Correct	Mark received	10/10	Time used	8	Times completed	2	Times submitted	2	Level	Medium	Question type	Single File Programming	Subject	Python
Topic	Exception Handling	Sub Topic	try except	Automatic marking	Apply										

Show testcases scores Show solution

Question No. 2

Single File Programming Question

Problem Statement

Write a program to read the Register number and Mobile Number of a student. Create user-defined exception and handle the following:

- If the Register Number does not contain exactly 9 characters in the specified format(2 numbers followed by 3 letters followed by 4 numbers) or if the Mobile Number does not contain exactly 10 characters, throw an `IllegalArgumentException`.
- If the Mobile Number contains any character other than a digit, raise a `NumberFormatException`.
- If the Register Number contains any character other than digits and alphabets, throw a `NoSuchElementException`.
- If they are valid, print the message "Valid" or else print its invalid message.

Input format :

The first line of the input consists of a string representing the Register number.
The second line of the input consists of a string representing the Mobile number.

Output format :

The output should display any one of the following messages:
If both numbers are valid, print "Valid".
If an exception is raised, print "Invalid with exception message:", followed by the specific exception message.

Refer to the sample output for the formatting specifications.

Code constraints :

In this scenario, the given test cases will fall under the following constraints:
The register number should have exactly 9 characters: 2 digits, 3 letters, and 4 digits.
The mobile number should have exactly 10 digits.

Sample test cases:

Input 1:

```
9876543210
9876543210
```

Output 1:

```
Valid
```

Input 2:

```
9876543210
9876543210
```

Output 2:

```
Invalid with exception message: Mobile Number should have exactly 10 characters.
```

Input 3:

```
9876543210
9876543210
```

Output 3:

```
Invalid with exception message: Register Number should have exactly 9 characters.
```

Input 4:

```
9876543210
9876543210
```

Output 4:

```
Invalid with exception message: Register Number should have the format: 2 numbers, 3 characters,
```

Input 5:

```
9876543210
9876543210
```

Output 5:

```
Invalid with exception message: Mobile Number could only contain digits.
```

All your code here:

Python 3.7

```
class IllegalArgumentException(Exception):
    """Exception raised for invalid length or format issues."""
    pass

class NumberFormatException(Exception):
    """Exception raised when mobile number contains non-digit characters."""
    pass

class NoSuchElementException(Exception):
    """Exception raised when register number contains invalid characters."""
    pass

def validate_register_number(register_number):
    """Validates the register number format."""
    if len(register_number) != 9:
        raise IllegalArgumentException("Register Number should have exactly 9 characters.")

    if not re.match(r"^\d{2}[A-Z]{3}\d{4}$", register_number):
        raise IllegalArgumentException("Register Number should have the format: 2 numbers, 3 characters, and 4 numbers.")

    if not register_number.isalnum():
        raise NoSuchElementException("Register Number should only contain digits.")
```

```

raise ValueError("Register Number should only contain digits and alphabets!")
else:
    validate_mobile_number(mobile_number)
    """Validates the mobile number format."""
    if len(mobile_number) > 10:
        raise IllegalArgumentException("Mobile Number should have exactly 10 characters!")
    if not mobile_number.isdigit():
        raise NumberFormatException("Mobile Number should only contain digits.")

```

Status: **Correct** - Marks obtained: 10/10 - Marks lost: 0 - Times compiled: 3 - Times submitted: 2 - Level: Head - Question type: Single File Programming - Subject: Python
Topic: **Exception Handling** - Sub Topic: User Defined Exception Handling - Previous Economy - Apply

Show Testcase source Show solution

Question No. 3

Single File Programming Question

Problem Statement:

In the enchanted world of Academia, you, the Academic Archivist, are bestowed with a magical scroll and a parchment to review the grades of copious students into a treasury of academic brilliance.

The mission is to craft a Python program that empowers faculty members to enter student grades for any two subjects; store these magical grades in a mystical file, and then, with a wave of your virtual hand, calculates the GPA to unveil the true essence of academic achievement.

Input Format:

The input format is a string representing the student's name, any two subjects, and corresponding grades.
After entering grades, they can type 'done' when prompted for the student's name.

Output Format:

The output should display the [average of grades] calculated GPA with a precision of two decimal places.
The magical grades will be stored in a mystical file named 'magical_grades.txt'.

Refer to the sample output for format specifications.

Code constraints:

The given test cases fall under the following constraints:

The grades should be from 0 to 100.

Sample test cases:

Input 1:

```

Alice
Math
95
English
89
done

```

Output 1:

```
61.50
```

Input 2:

```

Boris
Literature
78
Special
91
done

```

Output 2:

```
85.00
```

All your code here:

Python 3.8

```

# save_grades(student_details):
#     """Save grades to the mystical file."""
#     with open('magical_grades.txt', 'w') as file:
#         for student, subjects in student_details.items():
#             file.write(f'{student}:{subjects}\n')

# calculate_gpa(gpa):
#     """Calculate GPA with enhanced precision."""
#     return round((sum(gpa) / len(gpa)), 2)

# myglobal variables
student_details = {}

while True:
    student_name = input("Enter student name: ")
    if student_name.lower() == "done":
        break

    subjects = {}
    for _ in range(2):
        subject = input("Enter subject: ")
        grade = int(input("Enter grade: "))

        # Ensure grades are within the magical realm's limits
        if not 0 <= grade <= 100:
            print("Invalid grade! Please enter a number between 0 and 100!")
            continue

        subjects[subject] = grade

    student_details[student_name] = subjects
    gpa = calculate_gpa(list(subjects.values()))
    print(f"GPA: {gpa}")

```

Status: **Correct** - Marks obtained: 10/10 - Marks lost: 0 - Times compiled: 30 - Times submitted: 8 - Level: Head - Question type: Single File Programming - Subject: Programming - Topic: Python
Sub Topic: **File operation** - Previous Economy - Apply

Show testcase scores Show solution

Question No. 4

Single File Programming Question

Problem Statement

Write a program to obtain the start time and end time for the stage event show. If the user enters a different format other than specified, an exception occurs and the program is interrupted. To avoid this, handle the exception and prompt the user to enter the right format as specified.

- Start time and end time should be in the format "YYYY-MM-DD HHMMSS".
- If the input is in the above format, print the start time and end time.
- If the input does not follow the above format, print "Event time is not in the format".

Input format:

The first line of input consists of the start time of the event.

The second line of the input consists of the end time of the event.

Output format:

If the input is in the given format, print the start time and end time.

If the input does not follow the given format, print "Event time is not in the format".

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

The input format, YYYY-MM-DD HHMMSS.

Sample test cases:

Input 1:

```
2012-01-12 00:00:00
2012-01-12 10:10:10
```

Output 1:

```
2012-01-12 00:00:00
2012-01-12 10:10:10
```

Input 2:

```
2012-01-12 00:00:00
2012-01-12 10:00:
```

Output 2:

Event time is not in the format

Input 3:

```
2012-01-12 00:00:00
2012-01-31 10:10:10
```

Output 3:

Event time is not in the format

Input 4:

```
2012-01-12 10:00:00
2012-01-12 10:10:00
```

Output 4:

Event time is not in the format

Enter your code here:

```
#!/usr/bin/python
# exercise
# validate_datetime_format(date_str)
#   Validate if the given date string is in "YYYY-MM-DD HHMMSS" format.
#   If yes, validate_date = validate_datetime(date_str, "%Y-%m-%d %H:%M:%S")
#   return validate_date
#   else, ValueError
#   raise ValueError
# start_time = input().strip()
# end_time > start_time

# validate_start = validate_datetime_format(start_time)
# validate_end = validate_datetime_format(end_time)

# if validate_start < validate_end:
#     print(start_time)
#     print(end_time)
# else:
#     print("Event time is not in the format")
```

Filesize: 0B

Status: **Correct** | Last modified: 10/10 | Published: 8 | Times compiled: 3 | Times submitted: 2 | Level: Medium | Question type: Single file programming | Subject: Python

Show testcase scores Show solution

Student: kamal [r] - Email: 240701225@rejalahmied.edu.in - Test: EDC_Python_Week 5_TAH - Course: NeoColab_EDC_C523221_Python Programming

IP Address: 10.246.82.212 (200.40.44.6) 00:00:00:00:00:00
 Last login: 00:00:00

Total switches: 1
 Last start time: May 25 10:21:00 2024 (4h)

OS load: 0.00 0.00 0.00
 Last alarm time: May 25 10:21:00 (00:45:00)

Driver load: 0.00 0.00 0.00
 Max queue count: 1

summary Section

Edit Help

Coding (8)

Question No. 1

Single File Programming Question**Problem Statement:**

Peter maintains a student database and needs a program to add students, for each student, Alice inputs their ID and name. The program checks for duplicate IDs and ensures the database isn't full.

If a duplicate or full database is detected, an appropriate error message is displayed. Otherwise, the student is added, and a confirmation message is shown. The database has a maximum capacity of 20 students, and each student must have a unique ID.

Input format:

The first line contains an integer n , representing the number of students to be added to the school database.

The next n lines will contain two space-separated values, signifying the student's ID (integer) and the student's name (string).

Output format:

The output will depend on the actions performed in the code.

If a student is added to the database, the output will display "Student with ID [ID number] added to the database."

If there is an exception due to a duplicate student ID, the output will display "Exception caught. Note: Student ID already exists."

If there is an exception due to the database being full, the output will display "Exception caught. Note: Student database is full."

Refer to the sample outputs for the formatting specifications.

Code constraints:

The given test code will fail under the following constraints:

MAX_CAPACITY = 20;

1 <= MAX_CAPACITY

1 <= student_ID < 100

1 <= length of student's name < 100

Each student ID must be a positive integer

Sample Test Cases:**Input 1:**

```
3
30 Sam
31 Laiari
32 Eini
```

Output 1:

```
Student with ID 30 added to the database.
Student with ID 31 added to the database.
Student with ID 32 added to the database.
```

Input 2:

```
3
34 Abigail
35 Sandy
36 Dorothy
```

Output 2:

```
Student with ID 34 added to the database.
Student with ID 35 added to the database.
Exception caught. Error: Student ID already exists.
```

Input 3:

```
31
32 Sam
33 Jack
34 Kim
35 Jennifer
36 Dominic
37 Phillippe
38 Tanya
39 Esther
40 Daniel
41 Sophia
42 Emily
43 Olivia
44 Isabella
45 Victoria
46 Natalie
47 Daniel
48 Leah
49 Sarah
50 Emily
51 Daniel
52 Leah
53 Sarah
54 Emily
55 Natalie
56 Daniel
57 Leah
58 Sarah
59 Natalie
60 Daniel
61 Leah
62 Sarah
63 Daniel
64 Leah
65 Sarah
66 Natalie
67 Daniel
68 Leah
69 Sarah
70 Natalie
71 Daniel
72 Leah
73 Sarah
74 Daniel
75 Leah
76 Sarah
77 Natalie
78 Daniel
79 Leah
80 Sarah
81 Natalie
82 Daniel
83 Leah
84 Sarah
85 Natalie
86 Daniel
87 Leah
88 Sarah
89 Natalie
90 Daniel
91 Leah
92 Sarah
93 Daniel
94 Leah
95 Sarah
96 Natalie
97 Daniel
98 Leah
99 Sarah
100 Natalie
```

Output 3:

```
Student with ID 31 added to the database.
Student with ID 32 added to the database.
Student with ID 33 added to the database.
Student with ID 34 added to the database.
Student with ID 35 added to the database.
Student with ID 36 added to the database.
Student with ID 37 added to the database.
Student with ID 38 added to the database.
Student with ID 39 added to the database.
Student with ID 40 added to the database.
Student with ID 41 added to the database.
Student with ID 42 added to the database.
Student with ID 43 added to the database.
Student with ID 44 added to the database.
Student with ID 45 added to the database.
Student with ID 46 added to the database.
Student with ID 47 added to the database.
Student with ID 48 added to the database.
Student with ID 49 added to the database.
Student with ID 50 added to the database.
Student with ID 51 added to the database.
Student with ID 52 added to the database.
Student with ID 53 added to the database.
Student with ID 54 added to the database.
Student with ID 55 added to the database.
Student with ID 56 added to the database.
Student with ID 57 added to the database.
Student with ID 58 added to the database.
Student with ID 59 added to the database.
Student with ID 60 added to the database.
Student with ID 61 added to the database.
Student with ID 62 added to the database.
Student with ID 63 added to the database.
Student with ID 64 added to the database.
Student with ID 65 added to the database.
Student with ID 66 added to the database.
Student with ID 67 added to the database.
Student with ID 68 added to the database.
Student with ID 69 added to the database.
Student with ID 70 added to the database.
Student with ID 71 added to the database.
Student with ID 72 added to the database.
Student with ID 73 added to the database.
Student with ID 74 added to the database.
Student with ID 75 added to the database.
Student with ID 76 added to the database.
Student with ID 77 added to the database.
Student with ID 78 added to the database.
Student with ID 79 added to the database.
Student with ID 80 added to the database.
Student with ID 81 added to the database.
Student with ID 82 added to the database.
Student with ID 83 added to the database.
Student with ID 84 added to the database.
Student with ID 85 added to the database.
Student with ID 86 added to the database.
Student with ID 87 added to the database.
Student with ID 88 added to the database.
Student with ID 89 added to the database.
Student with ID 90 added to the database.
Student with ID 91 added to the database.
Student with ID 92 added to the database.
Student with ID 93 added to the database.
Student with ID 94 added to the database.
Student with ID 95 added to the database.
Student with ID 96 added to the database.
Student with ID 97 added to the database.
Student with ID 98 added to the database.
Student with ID 99 added to the database.
Student with ID 100 added to the database.
```

```
102. samip
20. sany
31. sali
31. craves
21. vangis
94. varonic
47. jenil.
33. pallini
```

Student with ID 30 added to the database.
Student with ID 15 added to the database.
Student with ID 11 added to the database.
Student with ID 311 added to the database.
Student with ID 95 added to the database.
Exception caught: Error: Student database is full.

Run your code here

```
MAX_CAPACITY = 30

i = int(input())
database = set()

for _ in range(i):
    try:
        entry = input().strip()
        if not entry:
            continue
        parts = entry.split(maxsplit=1)
        if len(parts) != 2:
            continue # ignore malformed line
        student_id_str, student_name = parts
        student_id = int(student_id_str)

        if len(database) == MAX_CAPACITY:
            raise Exception("Student database is full.")
        if student_id in database:
            raise Exception("Student ID already exists.")

        database.add(student_id)
        print(f"Student with ID {student_id} added to the database.")
    except Exception as e:
        print(f"Exception caught: {e}")
    break
```

Python... ▾

Status:	Correct	Mark obtained:	10/10	Time used:	2	Time compiled:	10	Time submitted:	8	Level:	Hard	Question type:	single file programming	submit	Python
---------	---------	----------------	-------	------------	---	----------------	----	-----------------	---	--------	------	----------------	-------------------------	--------	--------

Show testcode scores Show solution

Question No. 2

Single File Programming Question

Problem Statement:

John is a data analyst who often works with text files. He needs a program that can encrypt the contents of a text file and count the number of times a specific character appears in the file.

John wants a simple program that allows him to specify a file and a character to count within that file.

Input format:

The first line of input consists of the file's name to be analyzed.
The second line of the input consists of the string they want to write within the file.
The third line of the input consists of a character to count within the file.

Output format:

If the character is found, the output displays "The character % appears (N) times in the file" where % is the character and N is the count.
If the character does not appear in the file, the output displays "Character not found".

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:
The character is to be counted as a single character (alphabetic, numeric, or special character).
The length of the input string is 100.

Sample test cases:

Input 1:
text.txt
This is a text file to check the character count.
e

Output 1:
The character 'e' appears 5 times in the file.

Input 2:
sample.txt
lorem ipsum dolor sit amet, consectetur adipiscing elit.
i

Output 2:
The character 'i' appears 3 times in the file.

Input 3:
data.txt
Some random content with no special characters.
\$

Output 3:
Character not found in the file.

Input 4:

```
document.txt
This document contains multiple spaces and a character count.
```

Output 4:

The character ' ' appears 10 times in the file.

```
#!/usr/bin/python3

filename = input().strip()
text_to_write = input().lower()
char_to_count = input().lower()
with open(filename, 'w', encoding='utf-8') as file:
    file.write(text_to_write)
with open(filename, 'r', encoding='utf-8') as file:
    content = file.read()
    count = 0
    for i in content:
        if i==char_to_count:
            count+=1
if count > 0:
    print("The character \"{}\" appears {} times in the file.".format(char_to_count, count))
else:
    print("Character not found in the file.")
```

Status: **Correct** | Last updated: **10/10** | Submitted: **8** | Time compiler: **14** | Unsigned: **3** | Level: **Read** | Question type: **Single File Programming** | Subject: **Python** | Topic: **File Handling**

Show testcases scores Show solution

Question No. 3**Single File Programming Question****Problem Statement:**

Sheeta is playing with numbers. She wants to take a file containing a list of numbers, which she needs to find the average of those numbers. Write a program to read the numbers from the file, calculate the average, and display it.

File name: user_input.txt

Input format:

The input file will contain a single line of space-separated numbers (or a string). These numbers may be integers or decimals.

Output format:

If all inputs are valid numbers, the output should print "Average of the numbers is: 8.33" (where 8.33 is the computed average rounded to two decimal places).

If the input contains invalid data, print "Invalid data in the input".

Refer to the sample output for format specifications.

Code constraints:

The given test cases fall under the following constraints:

The file will have between 1 and 100 numbers ($1 \leq n \leq 100$).

Sample test cases:**Input 1:**

1 2 3 4 5

Output 1:

Average of the numbers is: 3.00

Input 2:

abc 1.1. xyz 2.2. ghi

Output 2:

Invalid data in the input.

Input 3:

5 7 3 6 5 9 38

Output 3:

Average of the numbers is: 10.00

#!/usr/bin/python3

```
filename = "user_input.txt"

try:
    # Read the lines from the file
    with open(filename, 'r', encoding='utf-8') as file:
        lines = file.readlines()
        tokens = []
        for line in lines:
            tokens += line.split()

    # Then converting all tokens to float
    numbers = [float(token) for token in tokens]
```

Python...

```
numbers = []
for token in tokens:
    try:
        number = float(token)
        numbers.append(number)
    except ValueError:
        print("Invalid data in the input!")
        break
else:
    # Only runs if the loop didn't break (all numbers are valid)
    avg = sum(numbers) / len(numbers)
    print(f"Average of the numbers is: {avg:.2f}")

except Description:
    print("Invalid data in the input!")

# = 1
s = input("Enter 0 or 1\n")
for i in s:
    if i.isdecimal():
        s = int(i)
        if s > 0:
            avg = 0.0
        else:
```

Status: **Partially solved** | Last submitted: **1.5/10** | Run time: **0** | Times compiled: **10** | Times submitted: **2** | Level: **Medium** | Reaction type: **Single file Programming** | Subject: **Programming**

Show testcase scores Show solution

Student: kamalraj - Email: 240701225@rejpalakshmi.edu.in - Test: EDC_Python_Week 5_000 - Course: NeoLab_REC_CS2221_Python Programming

IP Address: 240.47.4.255:542370-0b54ff2ba0
Test duration: 00:03:08Job switches: 1
Last start time: May 25, 2021 07:02 PMOffload: 0 hours
Last submit time: May 25, 2021 07:02 PMDriver test: 0 tests
Running count: 1

summary section

Edit Run

coding (0)

Question No. 1

single File Programming Question**Problem Statement:**

Sophie enjoys playing with words and wants to obtain the number of words in a sentence. She inputs a sentence, saves it to a file, and then reads it from the file to obtain the words.

Write a program to determine the number of words in the input sentence.

File Name: sentence_file.py**Input format:**

The input consists of a single line of text containing words separated by spaces.

Output format:

The output displays the count of words in the sentence.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

The length of the input string ≤ 1000 .

Sample test cases:**Input 1:**

Four Words In This Sentence

Output 1:

5

Input 2:

None NonSense NonSenseNonSense NonSenseNonSense

Output 2:

8

Input 3:**Output 3:**

0

All your code here:

Python... M

```
# You are using Python.
# Step 1: Get user input
sentence = input()

# Step 2: Write the sentence to the file
with open('sentence_file.txt', 'w') as file:
    file.write(sentence)

# Step 3: Read the sentence from the file
with open('sentence_file.txt', 'r') as file:
    content = file.read()

# Step 4: Count the words
# Split by whitespace and filter out any empty strings
words = content.split()
word_count = len(words)

# Step 5: Print the word count
print(word_count)
```

Status	Correct	Mark obtained	10/10	Submitted	8	Times compiled	3	Times submitted	1	Level	Easy	Question type	Single File Programming	Subject	Python	Topic	File Handling
--------	---------	---------------	-------	-----------	---	----------------	---	-----------------	---	-------	------	---------------	-------------------------	---------	--------	-------	---------------

Sub Topic: File Handling Bloom's Taxonomy: Apply

Show test case scores Show solution

Single File Programming Question**Problem Statement:**

Write a program that calculates the average of a list of integers. The program prompts the user to enter the length of the list (n) and each element of the list. It performs error handling to ensure that the length of the list is non-negative integer and that each input element is a numeric value.

Input format:

The first line of the input is an integer n , representing the length of the list as a positive integer.

The second line of the input consists of n elements of the list as an integer, separated by a new line.

Output format:

If the length of the list is not a positive integer or zero, the output displays "Error: The length of the list must be a non-negative integer."

If a non-numeric value is entered for the length of the list, the output displays "Error: You must enter a numeric value."

If a non-numeric value is entered for a list element, the output displays "Error: You must enter a numeric value."

If the inputs are valid, the program calculates and prints the average of the provided list of integers with two decimal places. "The average is: [average]"

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

$0 \leq n \leq 20$

The length of the list and each list element must be integers.

Sample test cases:**Input 1:**

```
4
3
2
1
```

Output 1:

```
Error: The length of the list must be a non-negative integer.
```

Input 2:

```
3
3
2
1
```

Output 2:

```
The average is: 2.00
```

Input 3:

```
3
4
2
1
```

Output 3:

```
Error: You must enter a numeric value.
```

Input 4:

```
4
3
2
1
```

Output 4:

```
Error: You must enter a numeric value.
```

Input 5:

```
4
3
2
1
```

Output 5:

```
Error: The length of the list must be a non-negative integer.
```

```
All your code here
# Python
# This code calculates list average.
n_input = input()
n = int(n_input)
if n <= 0:
    print("Error: The length of the list must be a non-negative integer!")
else:
    numbers = []
    for i in range(n):
        element = input()
        if element == "":
            print("Error: You must enter a numeric value!")
        else:
            number = float(element)
            numbers.append(number)
            if len(numbers) == n:
                print("Error: You must enter a numeric value!")
                break
    else:
        print("All inputs were valid, calculating average")
        avg = sum(numbers) / n
        print("The average is: ", avg)
```

Show testcases scores Show solution

Question No. 3

Single File Programming Question

Problem Statement:

In a voting system, a person must be at least 18 years old to be eligible to vote. If a user enters an age below 18, the system should raise a user-defined exception indicating that they are not eligible to vote.

Input format:

The input contains a positive integer representing age.

Output format:

If the age is less than 18, the output displays "Not eligible to vote". Otherwise, the output displays "Eligible to vote".

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

1 ≤ age ≤ 100

Sample test cases:

Input 1:

18

Output 1:

Eligible to vote

Input 2:

17

Output 2:

Not eligible to vote

All your code here:

```
# You are writing Python code
# Please use custom exception
class NotEligibleToVote(Exception):
    pass

# A function to check voting eligibility
def check_voting_eligibility(age):
    if age < 18:
        raise NotEligibleToVote("Not eligible to vote")
    else:
        print("Eligible to vote")

# Main block to take input and check eligibility
if __name__ == "__main__":
    age = int(input())
    check_voting_eligibility(age)
    except NotEligibleToVote as e:
        print(e)
```

Python... Run

Status: **Correct** | Last submitted: 10/10 | Submissions: 8 | Times compiled: 1 | Times submitted: 8 | Level: **Foxy** | Question-type: Single File Programming | Subject: Python | Topic: Exception Handling

Show testcase scores Show solution

Question No. 4

Single File Programming Question

Problem Statement:

Tara is a content manager who needs to perform case conversions for various pieces of text and save the results in a structured manner.

She requires a program to take a user's input string, save it in a file, and then retrieve and display the string in both **upper-case** and **lower-case** versions. Help her achieve this task efficiently.

File Name:

Input format:

The input consists of a single line containing a string provided by the user.

Output format:

The first line displays the original string read from the file in the format: "Original string: [original_string]"

The second line displays the upper-case version of the original string in the format: "[Upper-Case string] [upper_case_string]"

The third line displays the lower-case version of the original string in the format: [lower-case string] [lower_case_string].

Refer to the sample output for the formatting specifications.

Code constraints:

The input string can contain alphanumeric characters, spaces, and special symbols.
0 < length of the input string < 600

Sample test cases:

Input 1:

#SpecialSymbol123#

Output 1:

Original String: #SpecialSymbol123#
Upper-Case String: #SPECIALSYMBOL123#
Lower-Case String: #specialsymbol123#

Input 2:

lower_upper_space

Output 2:

Original String: lower_upper_space
Upper-Case String: LOWER_UPPER_SPACE
Lower-Case String: lower_upper_space

Input 3:

!@#\$%^&*()@#\$%^&*

Output 3:

Original String: !@#\$%^&*()@#\$%^&*
Upper-Case String: !@#\$%^&*()@#\$%^&*
Lower-Case String: !@#\$%^&*()@#\$%^&*

```
All your code here
1 // You are writing Python
2 # Write the FILE name
3 FILE_name = "test_file.txt"
4
5 # Get user input
6 user_input = input("Enter String")
7
8 # Validate user input
9 if len(user_input) > 300:
10     print("Error: The Input String must be between 3 and 300 characters.")
11     exit()
12
13 # Write the input string to the file
14 with open(FILE_name, "w") as f:
15     f.write(user_input)
16
17 # Read the string from the file
18 with open(FILE_name, "r") as f:
19     original_string = f.read().strip()
20
21 # Perform case conversion
22 upper_case_string = original_string.upper()
23 lower_case_string = original_string.lower()
24
25 # Display results
26 print(f"Original String: {original_string}")
27 print(f"Upper-Case String: {upper_case_string}")
28 print(f"Lower-Case String: {lower_case_string}")
```

State: **Correct** | Last updated: 10/10 | Submissions: 2 | Times compiled: 2 | Times submitted: 1 | Level: **Medium** | Question type: **Single File Programming** | Subject: **Programming**

Show Testcase scores Show solution

Question No. 5

Single File Programming Question

Problem Statement:

A retail store requires a program to calculate the total cost of purchasing a product based on its price and quantity. The program performs validation to ensure valid inputs and handles specific error conditions using exceptions.

- Price Validation: If the price is zero or less, raise a ValueError with the message: "Invalid Price".
- Quantity Validation: If the quantity is zero or less, raise a ValueError with the message: "Invalid Quantity".
- Cost Threshold: If the total cost exceeds 1000, raise a RuntimeError with the message: "Excessive Cost".

Input format:

The first line of input contains a double value representing the price of a product.

The second line contains an integer, representing the quantity of the product.

Output format:

If the calculation is successful, print the total cost rounded to one decimal place.

If the price is zero or less prints "Invalid Price".

If the quantity is zero or less prints "Invalid Quantity".

If the total cost exceeds 1000, prints "Excessive Cost".

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

10.0 : price : 100.0
14 quantity : 10000

Sample test cases:

Input 1:

20.0
5

Output 1:

100.0

Input 2:

10.0
6

Output 2:

Invalid Price

Input 3:

20.0
00

Output 3:

Excessive Data

Input 4:

20.0
4

Output 4:

Invalid Quantity

All your code here:

```
# This uses using try-except
def calculate_total_cost(price, quantity):
    try:
        if price <= 0:
            raise ValueError("Invalid Price")
        if quantity <= 0:
            raise ValueError("Invalid Quantity")
        total_cost = price * quantity
        if total_cost > 10000:
            raise ValueError("Excessive cost")
        print(f"Total cost: {total_cost}")
    except ValueError as e:
        print(e)
    except KeyboardInterrupt as e:
        print(e)

# Get user input
price = float(input())
quantity = int(input())
```

Python 3.9.1

Status: **Correct** - Marks obtained: 10/10 - Help used: 0 | Times compiled: 1 | Times submitted: 1 | level: **Medium** | Question type: Single File Programming | **Success**: Programming | Tags: Python

Last topic: **Description Handling** | Bloom taxonomy: **Apply**

Show testcase access Show solution

Student: Ismaili | Email id: 14070225@rajalakshmi.edu.in | Test: REC_Python_Week8_MQ | Course: NeoColab_REC_C523221_Python Programming

IP Address: 140.100.0.173 Date: 07/06/2023
Test Duration: 01:25:04Test Platform: —
Test Start Time: May 25, 2023 11:02:09 PMOS Used: Windows
Test End Time: May 25, 2023 11:44:09 PMBrowser Used: Firefox
Page View Count: 1

Summary

Section

Time

 MCQ [38]

Question No: 11

Multi Choice Type Question

Which of the following is true about the finally block in Python?

- The finally block is optional and does not have to be used.
- The finally block is executed only if an exception occurs.
- The finally block is always executed, regardless of whether an exception occurs or not.
- The finally block is executed only if no exception occurs.

Status: **Correct** Mark obtained: 1/1 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Exception Handling
 Show solution

Question No: 12

Multi Choice Type Question

How do you create a user-defined exception in Python?

- By creating a new class that inherits from the Exception class.
- By using the raise keyword.
- By writing a try block with custom logic.
- By defining a function and using it in try-except.

Status: **Correct** Mark obtained: 1/1 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Exception Handling
 Show solution

Question No: 13

Multi Choice Type Question

What will be the output of the following Python code?

```

1 # Predefined lines to simulate the file content
2 lines = [
3     "This is 1st line",
4     "This is 2nd line",
5     "This is 3rd line",
6     "This is 4th line",
7     "This is 5th line"
8 ]
9
10 print("Name of the file: foo.txt")
11
12 # Print the first 5 lines from the predefined list
13 for index in range(5):
14     line = lines[index]
15     print("Line No %d - %s" % (index + 1, line.strip()))

```

Compile Time Error:

displays output

Syntax Error

None of the mentioned options

Status: **Correct** | Mark obtained: **10** | Hints used: **0** | Level: **Medium** | Question type: **MCQ Single Correct** | Subject: **Python** | Topic: **File Handling** | Sub Topic: **File Handling**

Show solution

Question No: 14

Multi Choice Type Question

What is the correct way to raise an exception in Python?

`throwException()`

`exitException()`

`raise Exception()`

`catch Exception()`

Status: **Correct** | Mark obtained: **10** | Hints used: **0** | Level: **Easy** | Question type: **MCQ Single Correct** | Subject: **Programming** | Topic: **Python** | Sub Topic: **Exception Handling**

Show solution

Question No: 15

Multi Choice Type Question

What is the difference between `f1` and `f1.seek()`?

`f1` no difference

`f1.seek()` the pointer is initially placed at the beginning of the file and the pointer is at the end of the file.

`f1.seek()` depends on the operating system

`f1.seek()` the pointer is initially placed at the beginning of the file and the pointer is at the end of the file.

Status: **Correct** | Mark obtained: **10** | Hints used: **0** | Level: **Easy** | Question type: **MCQ Single Correct** | Subject: **Python** | Topic: **File Handling** | Sub Topic: **File Handling**

Show solution

Question No: 16

Multi Choice Type Question

Which of the following is true about `f1.seek(10)`?

`f1.seek(10)` Move file pointer ten characters behind from the current position

`f1.seek(10)` Move file pointer ten characters behind from the end of a file

Move file pointer two characters ahead from the current position

Move file pointer two characters ahead from the beginning of a file

Status: **Correct** Marks obtained: **10** Marks earned: **10** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling** Sub Topic: **File Handling**

Bloom's taxonomy: **Remember**

Show solution

Question No: 17

Multi Choice Type Question

What happens if no arguments are passed to the seek function?

file position remains unchanged

file position is set to the start of file

error

file position is set to the end of file

Status: **Wrong** Marks obtained: **0/10** Marks earned: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling** Sub Topic: **File Handling**

Bloom's taxonomy: **Remember**

Show solution

Question No: 18

Multi Choice Type Question

What is the output of the following code?

```
1 try:
2     x = 1 / 0
3 except ZeroDivisionError:
4     print("Caught division by zero error")
5 finally:
6     print("executed")
```

executed

ZeroDivisionError to be handled

ZeroDivisionError

Caught division by zero error
 Executed

Status: **Correct** Marks obtained: **10** Marks earned: **10** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Programming** Topic: **Python** Sub Topic: **Exception Handling**

Bloom's taxonomy: **Understand**

Show solution

Question No: 19

Multi Choice Type Question

What is the purpose of the except clause in Python?

To handle exceptions during code execution

To exit the program abruptly

To display the final output of the program

To define a custom exception

Status: **Correct** Score obtained: **10** Time taken: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **Input / File Handling** Sub Topic: **Exception Handling**

Remember

Show solution

Question No: 29

Multi Choice Type Question

Match the following:

- a) fseek(5) i) Move file pointer five characters behind from the current position
- b) fseek(-5) ii) Move file pointer to the end of a file
- c) fseek(0,2) iii) Move file pointer five characters ahead from the current position
- d) fseek(0) iv) Move file pointer to the beginning of a file

a-i, b-ii, c-iii, d-iv

O a-ii, b-iii, c-iv, d-i



a-i, b-ii, c-iii, d-iv

a-ii, b-iii, c-iv, d-i

Status: **Correct** Score obtained: **10** Time taken: **0** Level: **Medium** Question type: **MCQ single correct** Subject: **Python** Topic: **Input / File Handling** Sub Topic: **Exception Handling**

Remember

Student: Ismaili | Email id: 14070225@rajalakshmi.edu.in | Test: REC_Python_Week8_MQ | Course: NeoColab_REC_C523221_Python Programming

IP Address: 140.100.0.173 Date: 07/06/2020
Time: 07:25:01Top Layout: --
Test Run Date: May 25, 2020 11:02:09 PMOS: Mac OS X
Test Run Date: May 25, 2020 11:44:09 PMBrowser: Microsoft Edge
Test Run Date: May 25, 2020 11:44:09 PM

Summary

Section

7 hours

 MCQ (3)

+

Question No: 1

Multi Choice Type Question

What happens if an exception is not caught in the except clause?

- the program will display a traceback error and stop execution.
- raise Custom(exception)
- the program will exit automatically
- the exception is ignored and the program continues

Status: **Correct** Marks obtained: 0/1 Hint used: 0 Level: Easy Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Exception Handling

Blooms Taxonomy: Remember

 Show solution

Question No: 2

Multi Choice Type Question

What will be the output of the following Python code?

```
1: f = None
2: for i in range(5):
3:     with open("data.txt", "w") as f:
4:         if i > 3:
5:             break
6: print(f.closed)
```

Output:

- True
- compile time error

None of the mentioned options

Status: **Wrong** Marks obtained: 0/1 Hint used: 0 Level: Medium Question type: MCQ Single Correct Subject: Python Topic: File Handling Sub Topic: File Handling Show solution

Question No: 3

Multi Choice Type Question

How do you rename a file?

os.rename(existing_name, new_name)

- os.replace(existing_name, new_name)

```
fnname = 'new_name.txt'
```

```
os.rename(f, fnname)
```

Status: **Correct** Mark obtained: **10** Review count: **0** Level: **Medium** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling** Sub Topic: **File Handling**
Rooms Issuancy: **Remember**

Show solution

Question No: 4

Multi Choice Type Question

Fill in the blanks in the following code of writing data in binary file.

```
1 import _____(1)
2 rec=[]
3 while True:
4     rnum=int(input("Enter"))
5     rname=input("Enter")
6     temp=[rnum,rname]
7     rec.append(temp)
8     ch=input("Enter choice [y/N] ")
9     if ch.upper()=='N':
10         break
11 f=open("stud.dat","_____") (2)
12 _____.dump(rec,f) (3)
13 _____.close() (4)
```

(1).cPickle

(2).pickle(pickle)

(3).pickle(pickle)

(4).cPickle(pickle)

Status: **Correct** Mark obtained: **10** Review count: **0** Level: **Medium** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling** Sub Topic: **File Handling**
Rooms Issuancy: **Understand**

Show solution

Question No: 5

Multi Choice Type Question

Fill the code to In order to read file from the current position.

Assuming report file has following 3 lines; consider current file position is beginning of 2nd line.

Mark,25
John,31
Raj,20
Output:
[John,31]\n[Raj,20]\n]

```
1 f = open("report.txt", "r+")
2 _____(1)
3 print _____(2)
```

(1).read(0,1)
(2).readline()

(1).read(0,1)
(2).readline()

(1).read(0,1)
(2).read()

(1).read(0,1)
(2).read()

Status: **Correct** Mark obtained: **10** Review count: **0** Level: **Medium** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling** Sub Topic: **File Handling**
Rooms Issuancy: **Understand**

Show solution

Question No: 6

Multi Choice Type Question

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

Output:

Name of the file: abcd

```
3: fo = open('.....(1), "ab")  
3: print("Name of the file: ", .....(2))
```

- 0 'abcd'
 1 'abcde'

- 0 'abcd'
 1 fo.name()

- 0 exist
 1 fo.name()

- 0 exist
 1 fo.name

Status: **Correct** Mark obtained: **10** Estimated: **0** Level: **Medium** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling**

Sub topic: **Python directory and file management** Bloom's taxonomy: **Understand**

Show solution

Question No: 7

Multi Choice Type Question

Which clause is used to clean up resources, such as closing files in Python?

- else
 except
 try

- finally

Status: **Correct** Mark obtained: **10** Estimated: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Programming** Topic: **Python** Sub topic: **Exception Handling**

Show solution

Question No: 8

Multi Choice Type Question

What is the default value of reference_point in the following code?

```
file_object.seek(offset [,reference_point])
```

- 0 -1

- 1 null

- 2 garbage

Status: **Correct** Mark obtained: **10** Estimated: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **File Handling** Sub topic: **File Handling**

Show solution

Question No: 9

Multi Choice Type Question

What is the output of the following code?

```
1 try:
2     x = "Hello" + 5
3 except TypeError:
4     print("Type Error occurred")
5 finally:
6     print("this will always execute")
```

Executed

Type Error occurred
This will always execute

Type Error occurred

This will always execute

Status: **Correct** Marks obtained: 100 Marks used: 0 Level: **Easy** Question type: MCQ Single Correct Subject: **Programming** Topic: Python Sub Topic: Exception Handling Recommanded: Understand

Show solution

Question No: 10

Multi Choice Type Question

What is the output of the following code?

```
1 class MyError(Exception):
2     pass
3
4 try:
5     raise MyError("Something went wrong")
6 except MyError as e:
7     print(e)
```

Something went wrong

Type Error occurred

Executed

Nothing, the code will fail.

Status: **Correct** Marks obtained: 100 Marks used: 0 Level: **Easy** Question type: MCQ Single Correct Subject: **Programming** Topic: Python Sub Topic: Exception Handling Recommanded: Understand

Show solution

Student: kamalrj | Email: 248701225@jda.kthm.edu.in Test: R00_Python_Week 5_DY Course: NeoColab_R00_OG23201_Python Programming

IP Address: 10.24.8.252 (405484469@natwv0300-818) | Last Update: —

Last update: 60m ago

D1 User: 192.168.1.1

Last Update: May 20, 2020 (10:49 AM)

D2 User: 192.168.1.1

Last Update: May 22, 2020 (10:05 PM)

[Summary](#) [Details](#)[More](#)[Coding \(0\)](#)

Question No: 1

Single File Programming Question**Problem Statement:**

You own a store and keep track of item prices from two different suppliers using two separate dictionaries. We want to compare these prices to identify any differences. Your task is to write a program that calculates the absolute difference in price for items that are present in both dictionaries, for items that are unique to one dictionary (i.e., not present in the other), include them in the output dictionary with their largest price.

How to implement the above task using a dictionary?

Input Format:

The first line of input consists of an integer n_1 , representing the number of items in the first dictionary.

The next n_1 lines contain two integers:

The first line contains the item name (key_1), and

The second line contains the price ($value_1$).

The following consists of an integer n_2 , representing the number of items in the second dictionary.

The next n_2 lines contain two integers:

The first line contains the item name (key_2), and

The second line contains the price ($value_2$).

Output Format:

The output should display a dictionary that includes:

For items common to both dictionaries, the absolute difference between their price.

For items that are unique to one dictionary, the original price from that dictionary.

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

$1 \leq n_1 \leq 10$

$1 \leq n_2 \leq 10$

$1 \leq \text{value} \leq 50$

Sample Test Cases :**Input 1:**

```
1
A
10
B
20
C
30
D
40
E
50
```

Output 1:

```
{A: 0, B: 10, C: 30}
```

Input 2 :

```
1
A
10
B
20
C
30
D
40
E
50
F
60
G
70
H
80
I
90
J
100
```

Output 2:

```
{A: 10, B: 10, C: 30, D: 40, E: 50, F: 60}
```

Run your code here

```
# You are writing Python
# You have to calculate the differences and merge it into one
# compare periods
# A function for the first dictionary
def d1():
    #dict1 = {}
    dict1 = {}
    #for i in range(1, 11):
    for i in range(1, 11):
        item = input()
        price = int(input())
        dict1[item] = price
    #print("First period end item count", len(dict1))
    print("First period end item count", len(dict1))
    #return dict1
    return dict1
#dict2 = {}
dict2 = {}
#for i in range(1, 11):
for i in range(1, 11):
    item = input()
    price = int(input())
    dict2[item] = price
#print("Second period end item count", len(dict2))
print("Second period end item count", len(dict2))
#return dict2
return dict2
#dict3 = {}
dict3 = {}
#for item in dict1:
for item in dict1:
    if item in dict2:
        diff = abs(dict1[item] - dict2[item])
        dict3[item] = diff
    else:
        dict3[item] = dict1[item]
for item in dict2:
    if item not in dict1:
        dict3[item] = dict2[item]
#print("Final result", dict3)
print("Final result", dict3)
#if __name__ == "__main__":
if __name__ == "__main__":
    dict1 = d1()
    dict2 = d2()
    dict3 = merge(dict1, dict2)
    print("Final result", dict3)
if __name__ == "__main__":
    dict1 = d1()
    dict2 = d2()
    dict3 = merge(dict1, dict2)
    print("Final result", dict3)
```

Recent Submissions | My Submissions | Logout

Status: **Completed** | My Profile | Unsubscribed | 0 | Times compiled: 0 | Times submitted: 2 | Score: 100% | Question type: Single File Programming | Subject: programming | Topic: Python | Data Type: Int | Sub Type: Std

Show testcase scores Show solution

Question No. 2

Single File Programming Question

Problem Statement:

Billy 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. Billy needs to determine which books are still borrowed but has not returned, as well as those that were checked out and are now missing. Finally, she needs to find all unique books across both results.

Help! Billy is writing a program that performs the following operations on four sets of integers:

1. Compute the difference between the borrowed books (first set) and the returned books (second set).
2. Compute the difference between the added books (third set) and the missing books (fourth set).
3. 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.

Code constraints:

1 ≤ size_of_elements ≤ 500

Sample Test Cases:

Input 1:

```
1 2 3  
1 2 4  
1 4 2  
1 2 3
```

Output 1:

```
[3]  
[1]  
[1, 1]
```

Input 2:

```
1 2 3  
1 2  
1 2  
1 2 3
```

Output 2:

```
[1]  
[1]  
[1, 1]
```

Please code here:

```
# Import required modules  
from typing import List  
borrowed_books = set([int(i) for i in input().split()])  
returned_books = set([int(i) for i in input().split()])  
added_books = set([int(i) for i in input().split()])  
missing_books = set([int(i) for i in input().split()])  
  
# Compute set of borrowed books  
borrowed_set_returned = borrowed_books - returned_books, reverse=True  
added_new_missing = added_books - missing_books, reverse=True  
  
# Compute union of both results  
unique_books = sorted(list(borrowed_set_returned) + list(added_new_missing), reverse=True)  
  
# Print results  
print(len(borrowed_set_returned))  
print(len(added_new_missing))  
print(unique_books)
```

Recent Submissions | My Submissions | Logout

Status: **Completed** | My Profile | Unsubscribed | 0 | Times compiled: 0 | Times submitted: 2 | Score: 100% | Question type: Single File Programming | Subject: programming | Topic: Python | Data Type: Int | Sub Type: Std

Show testcase scores Show solution

Question No. 3

Single File Programming Question

Problem Statement:

Astronaut James is an engineer working on designing a new lunar 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_0, a_1, a_2 representing the coefficients of the quadratic equation.

Output Format:

The output displays:

If the discriminant is positive, display the two roots.

If the discriminant is zero, display the repeated root.

If the discriminant is negative, display the complex roots along with real and imaginary parts.

Refer to the sample output for formatting specifications.

Code constraints:

$N \leq 3$.

$|a|, |b|, |c| \leq 10$.

Sample Test Cases:

Input 1: 3 1 2 3	Output 1: $(-1.0, -3.0)$
Input 2: 1 1 1	Output 2: $(0.0, -4.0)$
Input 3: 3 1 2 3	Output 3: $\{(-1.0, -3.0), (-1.0, -2.0)\}$

All your code here

```
#include <iostream>
using namespace std;
int solve_quadratic(int N, int a[], int discriminant);
int discriminant(int a[], int b[], int c[]);
int main()
{
    int N;
    cout << "Enter the number of coefficients: ";
    cin >> N;
    int a[3];
    cout << "Enter the coefficients: ";
    for (int i = 0; i < N; i++)
        cin >> a[i];
    int discriminant = discriminant(a, a + N - 1, a + N);
    if (discriminant == 0)
        cout << "The roots are real and equal." << endl;
    else if (discriminant > 0)
        cout << "The roots are real and distinct." << endl;
    else
        cout << "The roots are complex conjugates." << endl;
    int roots[N];
    solve_quadratic(N, a, roots);
    cout << "The roots are: ";
    for (int i = 0; i < N; i++)
        cout << roots[i] << " ";
}

int discriminant(int a[], int b[], int c[])
{
    int discriminant = b[0] * b[0] - 4 * a[0] * c[0];
    return discriminant;
}

int solve_quadratic(int N, int a[], int roots[])
{
    int discriminant = discriminant(a, a + N - 1, a + N);
    if (discriminant == 0)
        roots[0] = -b[0] / (2 * a[0]);
    else
        roots[0] = (-b[0] + sqrt(discriminant)) / (2 * a[0]);
        roots[1] = (-b[0] - sqrt(discriminant)) / (2 * a[0]);
    return 2;
}
```

Python 3.7

Status: Partially correct | Last commit: 15/10 | Last Used: 0 | Pending Compute: 0 | Failed Submissions: 0 | Level: Hard | Constraint: Single File Programming | Editor: Python | Help | Datatype: auto | Topic: None

Show Testcase scores Show solution

Question No. 4

Single File Programming Question

Problem Statement:

Given two words in two different text files, that comprises few words to find common and unique letters. The user's program, that reads two words, A and B , and performs the following operations:

1. Sort the letters contained in both words in alphabetical order.
2. Print the letters that are unique to each word, in alphabetical order.
3. Determine the set of letters in one word that are a subset of the letters in the second word.
4. Check if there are no common letters between the two words and print the result as a Roman value.

Note: the program ignores case differences and treating leading spaces in the input words.

Your task is to help Samarth in implementing the same.

Input Format:

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

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

Output Format:

The first line of output should display the sorted letters common to both words, joined by comma.

The second line should display the sorted letters that are unique to each word, joined by a dot.

The third line should display a Roman value indicating if the set of letters in A is a superset of the set of letters in B .

The fourth line should display a Roman value indicating if there are no common letters between A and B .

Refer to the sample output for the formatting specifications.

Code constraints:

In this scenario, the given test cases will fall under the following constraints:

Don't use any code references, imports, or print statements.

1. Length of w1 and w2 is 10.

Sample test cases:

Input 1:

```
program  
func
```

Output 1:

```
[1, 2, 3]  
[1, 2, 3, 4, 5, 6, 7, 8, 9]  
True  
False
```

Input 2:

```
func func  
func
```

Output 2:

```
[1, 2, 3, 4, 5, 6]  
[1, 2]  
True  
False
```

Please code here:

```
#include <iostream.h>  
set<char> w1_set = set<char>(w1.begin(), w1.end());  
set<char> w2_set = set<char>(w2.begin(), w2.end());  
  
common_letters = w1_set.intersection(w2_set);  
unique_letters = w1_set.difference(w2_set);  
  
isCommonLetters = w1_set.issubset(w2_set);  
  
isUniqueLetters = w2_set.issubset(w1_set);  
  
cout << common_letters << unique_letters << isCommonLetters << isUniqueLetters;  
  
int main()  
{  
    string w1; string w2;  
  
    common_letters, unique_letters, isCommonLetters, isUniqueLetters = validateWords(w1, w2);  
  
    cout << common_letters << endl;  
    cout << unique_letters << endl;  
    cout << isCommonLetters << endl;  
    cout << isUniqueLetters << endl;  
}
```

System.out

Home Submit Recent My Submissions Help Discussion Test Questions Single File Programming Subject: Python Topic: Data Type Sort Topic Set

Show testcases scores Show solution

Student: kamalraj | Email id: 248701225@jda.kthm.edu.in Test: R00_Python_Week 5_M46 Course: NeoColab_R00_C03321_Python Programming

IP Address: 10.248.31.250 24/05/2019 14:46:09 IST (2020-05-24T14:46:09+05:30)

Tom-Delaney

—

D1 User

Windows

Last Test Date: May 18, 2020 | Status: U

Last Submit Date: May 22, 2020 | Status: P

Browser Used: Firefox

D1 User

Firefox

D1 User

Summary Details

Coding (W)

More

Question No: 1

Single File Programming Question**Problem Statement:**

Suppose you are given a list of event IDs representing consecutive days of an event. You need to group these IDs into consecutive sequences. For example, if the IDs 2, 6, and 9 appear consecutively, they should be grouped. Write a program that takes input 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 contains 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 events in a tuple format. Each group should be printed on a new line, and single-event should be displayed as a single-element tuple.

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases will use the following constraints:

1 ≤ n ≤ 30

1 ≤ tuple elements ≤ 25

Sample Test Cases :**Input 1:**1
2
3
4
5**Output 1:**

(1, 2, 3)

Input 2:6
7
8
9
10
11
12
13
14
15**Output 2:**(6, 7, 8, 9)
(10, 11, 12)**Input 3:**1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25**Output 3:**(1),
(2, 3, 4),
(5),
(6, 7, 8, 9, 10),
(11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25)**Run your code here**

Python 3.7.3

```
#!/usr/bin/python3
# group consecutive event IDs
event_ids = [int(x) for x in input().split()]
grouped_ids = []
temp_group = [event_ids[0]]
for i in range(1, len(event_ids)):
    if event_ids[i] == event_ids[i - 1]:
        temp_group.append(event_ids[i])
    else:
        grouped_ids.append(tuple(temp_group))
        temp_group = [event_ids[i]]
grouped_ids.append(tuple(temp_group)) # handle the last group
print(grouped_ids)

# Test case
n = int(input())
event_ids = [int(x) for x in input().split()]

# Present user output results
grouped_event_ids = group_consecutive_event_ids(event_ids)
if grouped_event_ids == grouped_ids:
    print("Correct")
else:
    print("Incorrect")
```

Question No. 2

Single File Programming Question

Problem Statement:

You need to write a dictionary that maps each integer from 1 to a given number n to its square. You will use this dictionary to quickly calculate the square of any number up to n.

Output must generate this dictionary based on the input file provided.

Input Format:

The input consists of one integer n, representing the highest number for which you want 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.

Code constraints:

The given test cases fall under the following constraints:

1 ≤ n ≤ 20

Sample Test Cases:

Input 1:

5

Output 1:

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

Input 2:

7

Output 2:

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49}

Fill your code here:

```
# If you are using Python
# Then do
n = int(input())
# Generate dictionary using dictionary comprehension
square_dict = {x: x*x for x in range(1, n+1)}
# Print the dictionary
print(square_dict)
```

Python 3.7.4

Status: **correct** | Last updated: **2023-06-14 11:45:23 UTC** | Time elapsed: **0** | Total accepted: **1** | Total solved: **1** | User: **Day** | Question type: **Single File Programming** | Subject: **Programming** | Topic: **Python** | Subtopic: **dictionary**

Show testcases scores Show solution

Question No. 3

Single File Programming Question

Problem Statement:

While working on a program to manipulate a set of integers, the program should draw inputs to perform the following operations:

1. Find the maximum value in the set.
2. find the minimum value in the set.
3. 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 is separated by a space.

The second line contains an integer m, representing the user's choice.

1 is find the maximum value

2 is find the minimum value

3 is remove a specific number from the set

If m is 0, the fourth line contains an integer n, which is the number to be removed from the set.

Output Format:

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

On output 1: Print the maximum value from the set.

On output 2: Print the minimum value from the set.

On output 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.

Code constraints:

The given test cases fall under the following constraints:

- 1 : Number of integers in the set <= 100
- 1 : Range (0, 100)

Sample test cases :

Input 1:

```
1 2 3 4 5
```

Input 2:

```
1 2 3 4 5 6 7
```

Input 3:

```
1 2 3 4 5  
6  
7
```

Input 4:

```
1 2 3 4 5  
6  
7
```

Input 5:

```
1 2 3 4  
5
```

My code here

```
# If the user enters symbols  
# then print error and end the loop  
initial_set = set(input().split(), repeat=False)  
  
# Read the user's choice  
ch = input()  
  
# If it is the original set or descending order  
if ch == "C":  
    print("Original set, reverse(most_left_index, most_right_index))")  
  
    # If the set is empty  
    if len(initial_set) == 0:  
        print("Initial set is empty")  
    else:  
        # Print and select the maximum value  
        print(max(initial_set))  
        # If the set has one element  
        if len(initial_set) == 1:  
            print("Initial set has one element")  
        # If the set has two elements  
        elif len(initial_set) == 2:  
            print("Initial set has two elements")  
            # Print the set in ascending order  
            print("Original set, reverse(most_left_index, most_right_index), 'P'")  
  
# If the user enters symbols  
# then print error and end the loop  
else:  
    print("Invalid choice")
```

Status: **Correct** | Most recent: **X90** | Unlocked: **0** | Once compiled: **1** | Once submitted: **1** | Last: **Java** | Languages: **Single File Programming** | Subject: **Python** | Type: **OctoType** | Sub Test: **90** |
Review Summary: **Apply**

Show testcase scores | Show solution

Question no. 4

Single File Programming Question

Problem Statement

Jasmin is writing 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 Jasmin 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.

Code constraints:

The given test cases fall under the following constraints:

- 1 : The elements of the set < 10000.

Sample test cases :

Input 1:

```
11 12 13 14
```

Input 2:

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

```
10 10 20 30 40 50
```

Input 3:

1 2 3 4 5

Output 3:

3 4 5 1 2

```
# You can edit this file.
# Don't forget to run it!
numbers = [1, 2, 3, 4, 5]
unique_numbers = []
for num in numbers:
    if num not in unique_numbers:
        unique_numbers.append(str(num)) # Insert str() to avoid type mismatch
unique_numbers.append(str(6)) # Insert 6 to avoid type mismatch
unique_numbers.append(str(7)) # Insert 7 to avoid type mismatch
unique_numbers.append(str(8)) # Insert 8 to avoid type mismatch
unique_numbers.append(str(9)) # Insert 9 to avoid type mismatch
unique_numbers.append(str(10)) # Insert 10 to avoid type mismatch
print("".join(unique_numbers))
```

Run Status: **Accepted** | Last updated: **10/09/2023** | Time elapsed: **0** | Views submitted: **1** | Level: **Medium** | Question type: **Single File Programming** | Status: **Python** | Topic: **Data Types** | Submissions: **1**

Review Score: **Apply**

Show testcases scores Show solution

Question No: 5

Single File Programming Question

Problem Statement

John is organizing a list of images. In a series of posts for his new project, she wants to create pairs of consecutive images from the list. The last image should be paired with None to complete the series. The pairing happens as follows: [image1, image2], [image3, image4], ..., [imageN, None].

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

Input Format:

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

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

Output Format:

The output displays a tuple containing pairs of consecutive images from the input. The last image in the tuple is paired with None.

Refer to the sample output for formatting specifications.

Code Constraints:

The given test cases fall under the following constraints:

$3 \leq n \leq 10$

$1 \leq \text{tuple elements} \leq 100$

Sample Test Cases:

Input 1:

5
1 2 3 4 5

Output 1:

(1, 2), (2, 3), (3, 4), (4, 5), (5, None)

Input 2:

4
1 2 3 4

Output 2:

(1, 2), (2, 3), (3, 4), (4, None)

You can edit this file.

```
# Don't forget to run it!
pairs = [(numbers[i], numbers[i+1]) for i in range(len(numbers)-1)]
pairs.append((numbers[-1], None)) # Add the last element of the list
print(pairs)

# Your Input
n = int(input())
numbers = list(map(int, input().split()))

# Create pairs and display result
print(create_pairs(numbers))
```

Run Status: **Accepted** | Last updated: **10/09/2023** | Time elapsed: **0** | Views submitted: **1** | Level: **Medium** | Question type: **Single File Programming** | Status: **Python** | Topic: **Data Types** | Submissions: **1**

Show test case scores Show solution

Question No. 8

Single File Programming Question

Problem Statement:

You want to create a dictionary that has the first n prime numbers, where each key represents the position of the prime number, and the value is the prime number itself.

Help function generates the dictionary based on the input and prints it.

Input Format:

The input consists of an integer n, representing the number of prime numbers. Then 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.

Code constraints:

The given test cases fall under the following constraints:

1 <= n <= 30

Sample test cases :

Input 1:

4

Output 1:

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

Input 2:

10

Output 2:

{1: 2, 2: 3, 3: 5, 4: 7, 5: 11, 6: 13, 7: 17, 8: 19, 9: 23, 10: 29, 11: 31, 12: 37, 13: 41, 14: 43, 15: 47}

Please code here

```
#!/usr/bin/env python3
# Check if a number is prime.
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# generate_prime_dict(),
# Returns a dictionary of the first n prime numbers.
prime = {}
count = 1
start = 2
while count <= n:
    if is_prime(start):
        prime[count] = start
        count += 1
    start += 1

print(prime)
# Main loop
n = int(input())
# Generate and print the dictionary
print(generate_prime_dict(n))
```

Show test case scores Show solution

Student: kamalraj | Email id: 248701225@pjms.edu.in Test: R00_Python_Week 5_000 Course: MeioClouds_R00_Python Programming

IP Address: 192.168.1.26
Last Accessed: 10/05/2023Test Submissions: 1
Last Test Run: May 10, 2023 (10:31 PM)Dif Used: 100%
Last Submit: May 10, 2023 (10:30 PM)Recent Used: Python
Last Used: May 10, 2023 (10:30 PM)

Summary Submissions

There

Coding (1)

Question No: 1

Single File Programming Question**Problem Statement:**

Given two lists of integers as input. Find the element-wise sum of the corresponding elements and then casting it into a tuple containing the sum values.
Write a program to solve Given problem with the test.

Example:

```
Input 1:
[1, 2, 3, 4]
[3, 5, 2, 1]
```

An element-wise sum of the said tuple: [4, 7, 5, 5]

Input Format:

The first line of input consists of a single integer n_1 , representing the length of the first list.
The second line of input consists of n_1 integers separated by commas, representing the elements of the first list.
The third line of input consists of n_2 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 formating specifications.

Code constraints:

In this section, the given two cases will follow the following constraints:
 $1 \leq n_1 \leq 10$
 >100 elements of the list ≤ 100

Sample Test Cases:**Input 1:**

```
4
1, 2, 3, 4
3, 5, 2, 1
```

Output 1:

```
(1, 7, 5, 5)
```

Input 2:

```
3
0, -1, 5
4, 5, 6
```

Output 2:

```
(3, 4, 11)
```

#Your code here

```
#if len(elementwise_sum([list1, list2])) == len(list1) + len(list2):
    for i in range(0, len(list1)):
        if len(list1[i]) != len(list2[i]):
            print("Error: The number of elements does not match the specified length")
        else:
            result = elementwise_sum([list1, list2])
            print(result)
else:
    print("Error: The number of elements does not match the specified length")
```

Menu: [Home](#) [Logout](#) [My Submissions](#) [R00](#) [Unsubmitted: 0](#) [Times compiled: 10](#) [Times submitted: 2](#) [Last Test](#) [Key](#) [Assignment Type](#) [Single File Programming](#) [Select Python](#) [Topic](#) [Data Type](#) [Run Test](#) [Topic](#)

Show Test Scores Show Solution

Question No: 2

Single File Programming Question

Problem Statement

In 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 workshop, the second set contains students who attended, and the third set contains students who dropped out.

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

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

Input Format:

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

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

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

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 union of the remaining student IDs that can be 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.

Code constraints

1 ≤ n ≤ elements ≤ 100

Sample Test Cases :

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

If you wish here

```
#include <iostream>
#include <set>
#include <algorithm>
#include <vector>
#include <map>
#include <string>
#include <assert.h>
using namespace std;

set<int> registered_and_attended(set<int> registered, set<int> attended) {
    set<int> attended_and_registered;
    for (int id : registered) {
        if (attended.find(id) != attended.end())
            attended_and_registered.insert(id);
    }
    return attended_and_registered;
}

set<int> attended_and_dropped_out(set<int> registered, set<int> attended) {
    set<int> attended_and_dropped_out;
    for (int id : registered) {
        if (attended.find(id) == attended.end())
            attended_and_dropped_out.insert(id);
    }
    return attended_and_dropped_out;
}

set<int> final_students(set<int> registered_and_attended, set<int> dropped_out) {
    set<int> final_students;
    for (int id : registered_and_attended) {
        if (dropped_out.find(id) == dropped_out.end())
            final_students.insert(id);
    }
    return final_students;
}
```

python_3.py

File

Edit

Insert

Format

Tools

Help

status connect Workstation 100% - released 0 Never compiled 1 times submitted 2 save say Question type Single File Programming Subject Python tags describe sub basic set

Run

Stop

Reset

Save

Save As

Apply

Show testcases scores Show solution

Question No. 3

Single File Programming Question

Problem Statement

Jones is managing a list of inventory items in a supermarket. EACH ITEM IS IDENTIFIED BY A NUMBER, WHERE THE FIRST NUMBER IS THE ITEM ID AND THE SECOND OVEREIGHT LINE OF QUANTITY AVAILABLE FOR THAT ITEM. JONES NEEDS TO TELL ME THE QUESTIONS THAT ARE ABOVE A CERTAIN THRESHOLD TO FIND ITEMS THAT HAVE A STOCK LEVEL BELOW THIS THRESHOLD.

Help Jones by writing a program to process these inputs. Read the quantities from all the available items, and display the results.

Notes

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 $(id, quantity)$, where id is an integer and the $quantity$ is an integer.

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, such quantity is exactly greater than the given threshold.

Refer to the sample output for formatting specifications.

Code Constraints

3 ≤ n ≤ 10

1 ≤ tuple elements ≤ 100

1 ≤ threshold ≤ 20

Sample Test Cases :

Input 1:

```
1  
1, 1, 1  
1, 1, 1  
1
```

Input 2:

```
1, 1, 1  
1, 1, 1  
1  
1
```

Output 1:

```
1, 1
```

Output 2:

```
1, 1
```

Run your code here

Python - 3.7

```
# Filter quantities less than threshold
filtered_quantities = []
for item_id, quantities in inventory:
    if sum(quantities) < threshold:
        filtered_quantities.append((item_id, quantities))
        print("Filtered quantities: ", filtered_quantities)

# Create a new inventory
new_inventory = {}
for item_id, quantities in filtered_quantities:
    total_qty = sum(quantities)
    new_inventory[item_id] = total_qty
threshold = int(input())
filtered_results = filter_quantities(new_inventory, threshold)
print("Filtered results: ", filtered_results)
```

Menu: [Solve](#) [Reset](#) [My Submissions](#) [Edit](#) [Unsubmit](#) [0](#) [Show compiled](#) [A](#) [Show submitted](#) [B](#) [Save](#) [Question type](#): Single File Programming [Submit](#) [Python](#) [Input](#) [DetectType](#) [Edit Topic](#) [Topic](#)

Show test case scores Show solution

Question No. 4

Single File Programming Question

Problem Statement:

You are managing the sales data for a new online shopping platform. You have a record of customer transactions where each customer's data includes their ID and a list of amounts spent on different items. You need to determine the total amount spent by each customer and identify the highest single expenditure for each customer.

You need to write a program that consumes three inputs and outputs them in the following:

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.](#)

Code constraints:

1 ≤ n ≤ 20

100 ≤ customerID ≤ 1000

100 ≤ expenditure ≤ 1000

Sample test cases:

Input 1:

```
3  
101 200 150 100  
102 50 10 100  
103 35 45 50 60 70
```

Output 1:

```
{101: [101, 150], 102: [101, 100], 103: [35, 70]}
```

Input 2:

```
3  
101 500 200 100 100  
102 40 90 120  
103 35 45 50 60 70
```

Output 2:

```
{101: [101, 500], 102: [101, 120], 103: [35, 70]}
```

Run your code here

Python - 3.7

```
# Analyse sales pattern, customer details
sales_summary = {}
for data in customer_data:
    customer_id = data[0]
    customer_id_element = [customer_id]
    sales_summary[customer_id_element] = []
    sales_summary[customer_id_element].append(data[1])
    total_expenditure = sum(data[1])
    max_expenditure = max(data[1])
    sales_summary[customer_id_element].append(total_expenditure)
    sales_summary[customer_id_element].append(max_expenditure)
    print(sales_summary)
n = int(input())
customer_data = []
for i in range(n):
    customer_data.append(list(map(int, input().split())))
    customer_data.append([])
```

Status: **Accepted** | My Submissions | **By Me** | Rejudged: 0 | Times compiled: 2 | Times submitted: 2 | Level: **Medium** | Question type: Single File Programming | Subject: Programming | Topic: Python | Sub-topic: **Dictionary**

Show test case scores Show solution

Question No. 8

Single File Programming Question

Problem Statement:

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

- i. Frequency of each product ID: A dictionary where the key is the product ID and the value is the number of times it appears.
- ii. Total number of unique product IDs.
- iii. 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
100
101
102
103
104
105 //product IDs
```

Output:

```
[{"100": 1, "101": 1, "102": 1}
Total Unique IDs: 3
Average Frequency: 2.00]
```

Explanation:

User's 6 indicates that you will enter 6 product IDs.
A dictionary is created to track the frequency of each product ID:
Input 100: Added with a frequency of 1.
Input 100: Added with a frequency of 1.
Input 101: Frequency of 100 increased to 1.
Input 102: Frequency of 100 increased to 1.
Input 103: Added with a frequency of 1.
Input 104: Frequency of 100 increased to 1.
Input 105: Frequency of 100 increased to 1.
The dictionary now contains 3 unique IDs: 100, 101, and 102.
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.

Code Constraints:

The given test cases fall under the following constraints:

1 ≤ n ≤ 100

100 ≤ product IDs ≤ 200

Sample Test Cases:

Input 1:

```
6
100
101
102
103
104
105
```

Output 1:

```
{"100": 1, "101": 1, "102": 1}
Total Unique IDs: 3
Average Frequency: 2.00
```

Input 2:

```
7
100
101
102
103
104
105
106
```

Output 2:

```
{"100": 1, "101": 1, "102": 1, "103": 1, "104": 1, "105": 1, "106": 1}
Total Unique IDs: 7
Average Frequency: 1.43
```

```
for product_id in product_ids:
    frequency_dict[product_id] = frequency_dict.get(product_id, 0) + 1
total_usage_ids = len(frequency_dict)
average_frequency = sum(list(frequency_dict.values())) / total_usage_ids
print(frequency_dict, total_usage_ids, average_frequency)
print(f"Total usage IDs: {total_usage_ids}")
print(f"Average frequency: {average_frequency:.4f}")
```

Status: **Solved** | Last updated: 2020-06-05 | Times solved: 4 | Times submitted: 2 | Level: **Medium** | Category: Single File Programming | Subject: Programming | Topic: Python | Edit Topic | Deliberate

Review difficulty: Apply

Show testcases Show solution

Student: Kamalraj | Email id: 248701225@pjstech.edu.in Test: R00_Python_Week5_MQ Course: NewColab_R00_0023221_Python Programming

IP Address: 192.168.250.247/49449/avivw/13500/818 | Test Details: —

Last Accessed: 05/20/2023, 10:10:49 AM | Last Submited: May 20, 2023, 10:10:49 AM

D1 User: 192.168.250.247/49449/avivw/13500/818 | Last Submited: May 20, 2023, 10:10:49 AM

Recent User: Pritika | D1 User: 192.168.250.247/49449/avivw/13500/818

[Summary](#) [Details](#)[More](#)[Help \(Q\)](#) [?](#)

Question No: 11

Multi Choice Type Question

Which of the following is a Python tuple?

 [1,2,3] [1,2,3] ()Status: **Correct** Mark obtained: 100% Retrieved: 5 Level: Medium Question type: MCQ Single Correct Subject: Python Topic: Datatype Author: hptek Status: Published Remember Show solution

Question No: 12

Multi Choice Type Question

What will be the output of the following program?

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

None of the mentioned options

invalid option

 [1,2,3,4]Status: **Correct** Mark obtained: 100% Retrieved: 5 Level: Medium Question type: MCQ Single Correct Subject: Python Topic: Datatype Author: hptek Status: Published Remember Show solution

Question No: 13

Multi Choice Type Question

Which of the statements about dictionary values is true?

 Values of a dictionary must be unique

More than one key can have the same value

Values of a dictionary can be a mixture of letters and numbers

The values of the dictionary can be accessed as dictionary[]

Status: **Correct** Mark obtained: 100% Retrieved: 5 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Datatype Author: hptek Status: Published Remember Show solution

Question No: 14

Multi Choice Type Question

What of the following statement is used to create an empty tuple?

 ()Status: **Correct** Mark obtained: 100% Retrieved: 5 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Datatype Author: hptek Status: Published Remember

Q1

II

II

Status: Correct Mark obtained: 0/0 Review count: 0 Level: Beginner Question type: MCQ Single Correct Subject: Python Topic: DataTypes Sub Topic: Sets Previous Next Save Answer Remember

Show solution

Question No. 18

Multi Choice Type Question

What will be the output for the following code?

```
1: M = {1, 2, 3, 4, 5}
2: N = {1, 2, 3, 4}
3: print(M > N)
```

Code

True

No output

False

Status: Correct Mark obtained: 0/0 Review count: 0 Level: Beginner Question type: MCQ Single Correct Subject: Python Topic: DataTypes Sub Topic: Sets Previous Next Save Answer Remember

Show solution

Question No. 19

Multi Choice Type Question

Set s = {1, 2, 4, 5} and t = {3, 5, 4, 6}. Then s & t = s | t, s ^ t = s & t, s - t = s - t.

```
s&t = {2, 3, 5, 6}
s-t = {1, 6}
s^t = {1, 2, 3, 5}
s|t = {1, 2, 3, 4, 5, 6}
```

s&t = {1, 4}
s-t = {1, 5}
s^t = {1, 3, 5, 6}
s|t = {1, 3, 4, 5, 6}

```
s&t = {1, 3, 5, 6}
s-t = {1, 2}
s^t = {1, 3, 5, 6}
s|t = {1, 2, 3, 5, 6}
```

s&t = {1, 4}
s-t = {1, 6}
s^t = {1, 5}
s|t = {1, 2, 3, 4, 5, 6}

Status: Wrong Mark obtained: 0/1 Review count: 0 Level: Very Question type: MCQ Single Correct Subject: Python Topic: Programming Sub Topic: Data Structures Previous Next Save Answer Remember

Question No. 20

Multi Choice Type Question

What is the result of print(type([])) in python?

list

None

True

Recommended learning Content : Data Types and Variables

Status: Correct Mark obtained: 0/0 Review count: 0 Level: Beginner Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Basics Previous Next Save Answer Remember

Show solution

Question No. 21

Multi Choice Type Question

Suppose $c = [1, 2, 4, 3]$, which of the following is incorrect?

print(c[0])

print([c[0]])

print(c[0])

Status: **Correct** Mark obtained: **10** Refreshed: **8** Level: **Easy** Question type: **MQ Single Correct** Subject: **Python** Topic: **DataTypes** Sub Topic: **Set** Home Learning: **None** Summary

Show solution

Question No: 18

Multi Choice Type Question

What is the output of the below Python code?

```
list1 = [1, 2, 3]
list2 = [5, 4, 2]
list3 = [10, 11, 12]
set1 = set(list2)
set2 = set(list3)
set1.update(set2)
set1.update(list5)
print(set1)
```

{1, 2, 3}

{1, 2, 3, 4, 5, 6, 10}

{1, 2, 3, 4, 5}

Type your answer

Status: **Correct** Mark obtained: **10** Refreshed: **8** Level: **Easy** Question type: **MQ Single Correct** Subject: **Python** Topic: **DataTypes** Sub Topic: **Set** Home Learning: **None** Summary: **Understand**

Question No: 23

Multi Choice Type Question

What is the output of the following?

```
int1 = [10, 20, 30, 40, 50]
int2 = [10, 20, 10, 20, 40, 80, 30, 60]
print(int1.isdisjoint(int2))
print(int1.isdisjoint(int1))
```

True

False

True

False

Type your answer

Status: **Correct** Mark obtained: **10** Refreshed: **8** Level: **Easy** Question type: **MQ Single Correct** Subject: **Python** Topic: **DataTypes** Sub Topic: **Set** Home Learning: **None** Summary: **Understand**

Show solution

Student: kamalraj | Email id: 248701225@jntuhkcm.edu.in Test: R00_Python_Week5_MQ Course: NewColab_R00_0023221_Python Programming

IP Address: 192.168.1.252 (49.49.144.99) and MAC: 00:0C:9E:00:00:0E
Last accessed: 10:00 AMTest Started: May 8, 2020 | 10:00 PM
Test Submitted: May 10, 2020 | 00:00:00Q1 Used: 0%
Used: 0.0000000000000001Q2 Used: 0%
Used: 0.0000000000000001

Summary Details

More

 Help Q1

Question No. 1

Multi Choice Type Question

Predict the output of the following Python program:

```
1. a,b,tuple_x = 1, 2, 3
2. init_tuple_p = (1, 2, 3)
3. a=c1=c2=tuple_x
4. a=c2=c1=tuple_x
5. print (set1 | set2)
6. print (init_tuple_x | init_tuple_x)
```

[1,2,3,4]
[1,2,3,0]

[1,2]
[3,4]

 Task 1: unsupported operand type Task 1: unsupported operand type

Status: Wrong | Marked correct: 0/1 | Accessed: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: datatype | Sub topic: INT | Difficulty level: Understand

Question No. 2

Multi Choice Type Question

What will be the output?

```
1. print(a[0],a[1],a[2])
2. print(b[0],b[1])
```

[A,B,C]

[B,C,D]

 [A,B,C]

[B,C,D]

Status: Correct | Marked correct: 1/1 | Accessed: 0 | Level: Medium | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub topic: Dictionaries | Difficulty level: Understand

 Show solution

Question No. 3

Multi Choice Type Question

What is the output of the following code?

```
1. print("1","2","3")
2. b= dict(c=c.values(),c.keys())
3. print(b)
4.
```

[None of the above]

An exception is thrown

[None of the above]

 [None of the above]

Status: Wrong | Marked correct: 0/1 | Accessed: 0 | Level: Hard | Question type: MCQ Single Correct | Subject: Python | Topic: Basic Python | Sub topic: Python Dictionaries | Difficulty level: Analyse

Question No. 4

Multi Choice Type Question

What will be the output for the following code?

```
1: a=[1,2,3]
2: b=[4,5,6]
3: c=a+b
4:
5: print(c)
6: print(type(c))
```

None of the mentioned options

{[1,2,3],[4,5]}

{[1,2,3],[4,5]} ✓

{[1,2,3][4,5]}

Status: Correct Mark assigned: 0/1 Released: 0 Level: Medium Question type: MCQ Single Correct Subject: Python Topic: Data type Subtopic: Type Status: Inactive Dictionary: Understand

Show solution

Question No. 6

Multi Choice Type Question

What is the following not true about dictionary keys?

When declared by user defined, the key assignment value

Keys must be integers ✓

Keys must be immutable

Multi-level keys are not allowed ✓

Status: Wrong Mark assigned: 0/1 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data type Subtopic: Dictionary Status: Inactive Dictionary: Understand

Show solution

Question No. 8

Multi Choice Type Question

What will be the output of the following code?

```
1: a=[1,2,3,4]
2: print(*a,sep=,)
```

?

9 ✓

The method `str().__format__()` used for `format()`

Too many arguments for `format()` method

Status: Correct Mark assigned: 0/1 Released: 0 Level: Medium Question type: MCQ Single Correct Subject: Python Topic: Data type Subtopic: Type Status: Inactive Dictionary: Understand

Show solution

Question No. 7

Multi Choice Type Question

What is the code in order to get the following output.

```
output:
Topic: 1,2,3
And value: 4
1: t=[1,2]
2: print("Topic",*,t)
3: print("Value",*,4)
4: print("The value",*,_____)

1:[1,2]
2:print()

1:[1,2]
2:print()

1:[1,2]
2:print()
```

print(*t,sep=,) ✓

[1,2]

Question No. 5

Mult Choice Type Question

What is the output of the following code?

- 1: `a=[1,2,(3,4,5)]
b=[1,2,(1,4)]
print(a+b)`

Error: + operator is not valid for tuples

 2: `a=[1,2,(3,4,5)]
b=[1,2,(1,4)]
print(a+b)`

Error: + operator is valid for tuples but not if there are extra spaces

True

Question No. 6

Mult Choice Type QuestionIf 'd' is a dictionary with some key-value pairs, what does `d.popitem()` do?

Removes all the keys from dict

 3: Removes an arbitrary element

Removes the key-value pair for the key given as an argument

Removes method for dictionary

 Show solution

Question No. 12

Mult Choice Type Question

What is the output of the following code?

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

[1: A, 2: B, 3: C]

Error: copy() method doesn't work for dictionaries

'None' is printed

 2: `(1, 'A', 2, 'B')` Show solution

Student: kamalraj | Email id: 248701225@pj.dakshin.edu.in Test: R00_Python_Week 4_OY Course: NeoColab_BLC_OG2021_Python Programming

File Version: 248701225@pj.dakshin.edu.in|Last Modified: 10-07-2023 10:25 AM

Test Version: 100% | Last Modified: 10-07-2023 10:25 AM

Build Used: Windows | Last Built: 10-07-2023 10:25 AM

Review Used: Python | Last Review: 10-07-2023 10:25 AM

Summary | Details

Edit | Close

Coding (0) |

Question No. 1

Single File Programming Question**Problem Statement:**

Arvind is developing a password strength checker for his website. He wants the user to consider the length and the diversity of characters used in the password. A strong password should have length and include a mix of character types (uppercase, lowercase, digits and special symbols).

The user wants the feedback to be user-friendly, so he wants to include the actual password in the output message from the password checker using Python's built-in string methods.

Character Types Considered:

- Lowercase letters [a-z]
- Uppercase letters [A-Z]
- Digits [0-9]
- Special characters (non-alphanumeric, e.g., @, !, %, #)

Input Format:

The input consists of a single string representing the user's password.

Output Format:

The program prints the strength of the password in this format:

If the password length is 8 characters or fewer OR if 2 of the 4 character types, the output prints "password is Weak".

If password length is 9 and at least 2 character types, the output prints "password is Moderate".

If password length is 10 and all 4 character types present, the output prints "password is Strong".

Refer to the sample output for formatting specifications.

Code Constraints:

You given test cases fall under the following conditions:

The password can be any combination of characters (letters, numbers, special symbols).

Sample Test Cases:**Input 1:**

password123

Output 1:

password123 is Moderate

Input 2:

123

Output 2:

123 is Weak

Input 3:

HelloWorld123

Output 3:

HelloWorld123 is Strong

ANSWER:

Test 1 | Run | Submit | Resubmit | Download

Edit your code here:

```
# check_password_strength(password):
    length = len(password)
    has_lowercase = any([c.islower() for c in password])
    has_uppercase = any([c.isupper() for c in password])
    has_digit = any([c.isdigit() for c in password])
    has_special = any([c in '@!%#' for c in password])
    char_types = [has_lowercase, has_uppercase, has_digit, has_special]
    if length < 8 or not char_types[0]:
        print("Weak")
    elif length >= 8 and length <= 10 and char_types == 3:
        print("Moderate")
    elif length >= 10 and char_types == 4:
        print("Strong")
    else:
        print("Unknown")

password = input()
strength = check_password_strength(password)
print(f'{password} is {strength}' + " " + str(strength))
```

Status - Success | Mark submitted | R00 | Last saved: 0 | Unsubmitted | 0 | Unsubmitted | 1 | level: Medium | Question type: Single File Programming | Subject: Programming | Test: Python | Sub Test: Function

Review history | Apply

Show test case scores | Show solution:

Question No: 3

single File Programming Question

Problem Statement:

You are tasked with designing a shipping cost calculator program that calculates the shipping cost for packages based on their weight and destination. The program utilizes different shipping rates for domestic, international, and remote destinations. The rates for each destination type are provided as global constants.

Constant Values:

```
DOMESTIC_RATE = 5.0  
INTERNATIONAL_RATE = 10.0  
REMOTE_RATE = 15.0
```

Function Signature: calculate_shipping(weight, destination):

```
formula: shipping_cost = weight * destination_rate
```

Input Format:

The first line of the input consists of a float representing the weight of the package.

The second line consists of a string representing the destination (Domestic or International or Remote).

Output Format:

The program outputs one of the following:

If the input is valid and the destination is Domestic, the output should consist of a single line stating the calculated shipping cost for the given weight and destination in the format: "Shipping cost to [destination] for a [weight] kg package".

If calculated cost is less than decimal places,

If the input weight is not a positive float, print "Invalid weight. Weight must be greater than 0."

If the input destination is not one of the valid options, print "Invalid destination".

Refer to the sample output for the formatting specifications.

Code constraints :

```
1.0 <= weight <= 100.0.  
Lowest constraint (0.0).  
destination is case sensitive.
```

Sample test cases :

Input 1:

```
1.0  
Domestic
```

Output 1:

```
Shipping cost to Domestic for a 1.0 kg package: $5.00
```

Input 2:

```
1.0  
International
```

Output 2:

```
Shipping cost to International for a 1.0 kg package: $10.00
```

Input 3:

```
1.0  
Remote
```

Output 3:

```
Invalid output - weight must be greater than 0.
```

Input 4:

```
1.0  
invalid
```

Output 4:

```
Invalid destination.
```

Editor Interface:



Edit your code here:

```
DOMESTIC_RATE = 5.0  
INTERNATIONAL_RATE = 10.0  
REMOTE_RATE = 15.0  
def calculate_shipping(weight, destination):  
    if weight < 0:  
        print("Invalid weight. Weight must be greater than 0.")  
        return None  
  
    if destination == "Domestic":  
        rate = DOMESTIC_RATE  
    elif destination == "International":  
        rate = INTERNATIONAL_RATE  
    elif destination == "Remote":  
        rate = REMOTE_RATE  
  
    else:  
        print("Invalid destination.")  
        return None  
    shipping_cost = rate * weight  
    print(shipping_cost)  
  
    weight = float(input())  
    destination = input()  
  
    shipping_cost = calculate_shipping(weight, destination)
```

Footer Output:

```
1.0  
Shipping cost to Domestic for a 1.0 kg package: $5.00
```

Status: **Correct** | Mark obtained: **10/10** | Time used: **0** | Times compiled: **10** | Times submitted: **1** | Level: **Hard** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Basic Python** | Back Topic | **python solutions**

Show testable scores Show solution

Question No. 3

Single File Programming Question

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 the 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 n in the format: "The smallest positive divisor of $|n|$ is [smallest divisor]."

Refer to the sample output for the test harness.

Code constraints:

The given test cases fall under the following constraints:
 $1 \leq n \leq 10000$

Sample test cases:

Input:	Output 1:
12	The smallest positive divisor of 12 is 2.

Input 2:	Output 2:
1	The smallest positive divisor of 1 is 1.

Write your code here

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    if (n == 1) {
        cout << "The smallest positive divisor of " << n << " is 1." << endl;
    } else {
        int minDivisor = 2;
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                minDivisor = i;
                break;
            }
        }
        cout << "The smallest positive divisor of " << n << " is " << minDivisor << endl;
    }
}
```

Status: **Correct** | Mark obtained: **10/10** | Time used: **0** | Times compiled: **4** | Times submitted: **1** | Level: **Medium** | Question type: **Single File Programming** | Subject: **Programming** | Topic: **Python** | Back Topic | **python solutions**

Show testable scores Show solution

Question No. 4

Single File Programming Question

Problem Statement:

Imagine you are tasked with developing a function for calculating the total cost of an item after applying taxes. 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:

Your Cost: [item cost]
Sales Tax: [sales tax rate]
Total Cost: [total cost]

Total cost followed by the calculated cost after applying the state tax formatted to two decimal places.

Refer to the sample output for formatting specifications.

Code constraints

0 ≤ item_cost ≤ 10³

Sample test cases

Input 1:

10.00

Output 1:

Item Cost: 10.00
Sales Tax Rate: 8.00
Total Cost: \$11.00

Input 2:

100.00

Output 2:

Item Cost: 100.00
Sales Tax Rate: 8.00
Total Cost: \$108.00

Header snippet

```
def calculate_total(item_cost):
    sales_tax_rate = 8.00
    tax = item_cost * SALES_TAX_RATE
    total = item_cost + tax
    return total

# Example
item_cost = float(input("Enter item cost: "))

# Calculation
final_cost = calculate_total(item_cost)

print(f"Final cost: ${final_cost:.2f}")
```

Footer snippet

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

Submit Cancel Max statement 8/10 Previous Next Home < Back Home > Forward < > Print Copy Clear All Type Single File Programming Subject Python Topic Past Due Sub Topics Results

Recent Activity: None

Show Testcase Scores Show Solution

Student: kamalraj | Email id: 248701225@pjmskhm.edu.in Test: R00_Python_Week 4_M4E_Updated Course: NeoCodez_R00_OG23231_Python Programming

① IP Address: 2405:fb01:0:102:1020:473:ff00:4000
② TestDuration: 00:00:00.000000000③ TestDuration: 00:00:00.000000000
④ TestDuration: 00:00:00.000000000⑤ DiffUser: 1000000
⑥ TestDuration: 00:00:00.000000000⑦ Recent User: Python
⑧ TestDuration: 00:00:00.000000000[Summary](#) [Statistics](#)[More](#)[Coding \(W\)](#)

Question No: 1

Single File Programming Question**Problem Statement:**

ABC services as a digital marketing company, where they analyzes large datasets. One day, user is faced with processing customer transaction, which are long numeric sequences.

To simplify the task, ABC needs to calculate the rightmost digit sum. This digit sum is obtained by repeatedly summing the digits of a number until a single digit remains.

Hence, ABC wants to program that needs to implement this function, calculate its digit root and prints the result using a loop-based approach.

For example, the sum of the digits of 555555 is $5 + 5 + 5 + 5 + 5 = 25$, then $2 + 5 = 7$, which is the digit root.

Function prototype: int digital_root(sum)

Input Format:

The input consists of an integer n.

Output Format:

The output prints an integer representing the sum of digits for a given number until a single digit is obtained.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

1 ≤ n ≤ 10⁹.

Sample Test Cases:

Input 1:	Output 1:
123456	6
Input 2:	Output 2:
12345	1
Input 3:	Output 3:
9999	8

Header Snippet:

```
from sys import stdin
```

```
def digital_root(n):
    while n > 10:
        sum = 0
        for i in str(n):
            digit = int(i)
            sum += digit
        n = sum
    return n
```

Footer Snippet:

```
print(digital_root(n))
```

Assignment: **Answers**

Show testcase scores Show solution

Question No. 3

Single File Programming Question

Problem Statement:

You have to develop a program to find consecutive digits in customer's account number. This has to generate unique verification codes for account numbers. The provided program has taken an account number as input and outputs the sum of its digits.

Note: You have to complete this task.

Function specifications: `def sum_digits(num)`

Input Format:

The input consists of an integer, representing the customer's account number.

Output Format:

The output prints an integer representing the sum of the digits of the account number.

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

$0 \leq num \leq 10^9$

Sample test cases:

Input 1:	Output 1:
123456	21
9	9

Header Output:

```
from sys import argv
```

```
if __name__ == "__main__":
    sum_digits = 0
    num = int(argv[1])
    while num > 0:
        digit = num % 10
        sum_of_digits += digit
        num = num // 10
    print(sum_of_digits)
```

Footer snippet:

```
from sys import argv
```

Assignment: **Answers**

Show testcase scores Show solution

Question No. 3

Single File Programming Question

Problem Statement

This is designing of compressing application that receives file handle from user efficiently. To optimize storage and transmission, the plan is implementation of lossless compression feature that replaces consecutive repeated characters with their character followed by its count, while leaving non-repeated character unchanged.

Goal: Create a recursive function to achieve this compression without altering the original message's meaning.

Function Specification: def compress_string(text)

Input Format:

The input consists of a single line containing the string to be compressed.

Output Format:

The output consists of a single line containing the compressed string.

Refer to the sample output for the formatting specifications.

Code constraints:

In this scenario, the given test cases will fall under the following constraints:

1. The length of the input string is 100.

The input string consists of lowercase and uppercase letters only.

Sample Test Cases:

Input	Output 1	Output 2	Output 3
aaabbbcc	aaabbcc	aaabbcc	aaabbcc
aaabbbccaa	aaabbccaa	aaabbccaa	aaabbccaa
aaabbbccaaab	aaabbccaaab	aaabbccaaab	aaabbccaaab

My code here

```
#include <iostream>
#include <string>
using namespace std;

string compress_string(string text) {
    if (text.length() == 0)
        return "";
    char compressed = text[0];
    int count = 1;
    for (int i = 1; i < text.length(); i++) {
        if (text[i] == compressed) {
            count++;
        } else {
            compressed = text[i];
            count = 1;
        }
        if (count > 1) {
            compressed += to_string(count);
        }
    }
    return compressed + compress_string(text.substr(count));
}
```

Status: **Accepted** | Last submitted: 10/09 | Time used: 0 | Time compiled: 0 | Errors detected: 0 | Level: Hard | Question type: Single File Programming | Subject: Algorithms | Topic: Recursion | R.A Tools: Recursion

Show testcases scores Show solution

Question No. 4

Single File Programming Question

Problem Statement

Create a Python program to monitor two sensors. In a greenhouse using two sensors. Calculate and display the absolute temperature difference between the two sensor readings to ensure proper temperature control. Note: Use the abs() built-in function.

Input Format:

The first line consists of a floating-point number representing the temperature reading from Sensor 1. The second line consists of a floating-point number representing the temperature reading from Sensor 2.

Output Format:

The output displays the absolute temperature difference between Sensor 1 and Sensor 2, rounded to four decimal places.

Refer to the sample output for the least format.

Code constraints:

The given test cases fall under the following constraints:

-100.0 ≤ temperature ≤ 100.0

Sample Test Cases:

Input	Output 1
35.2 35.7	Temperature difference: 0.5000

Input 2:

7.0
23.0

Output 2:

Temperature difference: 16.00 °C

Analyze

Run

Run your code here

Python 3.6

```
# calculate_temperature_difference(reading1, reading2):
    difference = abs(reading1 - reading2)
    sensor1_reading = float(input())
    sensor2_reading = float(input())
    temperature_difference = calculate_temperature_difference(sensor1_reading, sensor2_reading)
    print("Temperature difference: " + str(temperature_difference) + " °C")
```

Score: **Correct** | User submitted: 10/10 | Time used: 0 | Times compiled: 4 | Times submitted: 1 | Level: Medium | Category: Single File Programming | Subject: Programming | Type: Python | Sub-Type: Functions

Mark as complete

Show test case scores Show solution

Question No. 5

Single File Programming Question

Problem Statement

Ivan is writing an algorithm to determine how many of them are divisible by 3 and how many are divisible by 5. He decided to use formula batches to find and count the numbers based on their divisibility.

Write a program that takes a list of integers, calculates how many numbers are divisible by 3 and how many are divisible by 5, and then prints the results.

Additionally, the program should calculate the sum of all numbers divisible by 3 and divisible by 5 separately.

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 should print the count of numbers divisible by 3.

The second line should print the count of numbers divisible by 5.

The third line should print the sum of numbers divisible by 3.

The fourth line should print the sum of numbers divisible by 5 separately.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases will satisfy the following specifications:

1 ≤ n ≤ 100

1 ≤ number ≤ 100

Sample Test Cases:

Input:

9
1 3 6 18 21 30

Output 1:

3
1
24
54

Input 2:

9
1 2 3 4 13 17 19 23 29

Output 2:

0
0
0
0

Code Size: 1000.00 B

Analyze

Run

Run your code here

Python 3.6

```
# solution()
n = int(input())
numbers_str = input().split()
numbers = [int(x) for x in numbers_str]
```

```

is_variable_by_3 = lambda x: x % 3 == 0
is_variable_by_5 = lambda x: x % 5 == 0

count_divisible_by_3 = len([x for x in divisible_by_3_numbers])
count_divisible_by_5 = len([x for x in divisible_by_5_numbers])
sum_divisible_by_3 = sum([x for x in divisible_by_3_numbers])
sum_divisible_by_5 = sum([x for x in divisible_by_5_numbers])

print(count_divisible_by_3)
print(count_divisible_by_5)
print(sum(divisible_by_3))
print(sum(divisible_by_5))

if __name__ == "__main__":
    solve()

```

status: **Accepted** | last updated: **4 days ago** | time elapsed: **0** | time submitted: **2 days ago** | views: **Medium** | owner type: **Single File Programming** | subject: **Programming** | topic: **Python** | last topic: **Function**

Show testcases scores Show solution

Question No. 6

Single File Programming Question

Problem Statement

youman wants to create a program to calculate a person's BMI (Body Mass Index) based on their weight in kilograms and height in meters. The formula is $BMI = \frac{weight}{height^2}$. You need to write a program to calculate the BMI.

Your program should take user input for weight and height; calculate the BMI and display the result.

Function Signature: calculate_bmi(weight, height)

Returns: float = calculate_bmi()

Input Format :

The first line of input consists of a positive floating-point number, the person's weight in kilograms.

The second line of input consists of a positive floating-point number, the person's height in meters.

Output Format :

The output displays "Your BMI is: [val]" followed by a float value representing the calculated bmi, rounded off two decimal points.

Refer to the sample output for the formatting specifications.

Code constraints :

The test cases will fall under the following constraints:

1.0 < weight < 100.0

0.5 < height < 2.5

Sample test cases :

Input 1:

```
70.0
1.75
```

Output 1:

```
Your BMI is: 21.88
```

Input 2:

```
60.0
1.65
```

Output 2:

```
Your BMI is: 23.90
```

Header imports

```
weight = float(input())
height = float(input())
```

All you code here

```

def calculate_bmi(weight, height):
    if height <= 0:
        print("Height cannot be zero or negative.")
    elif weight <= 0:
        print("Weight cannot be zero or negative.")
    else:
        bmi = weight / (height * height)
        bmi_result = calculate_bmi(weight, height)
        print(f"Your BMI is: {bmi_result}")

```

Footer Dispal

calculate_imperfection_start()

Status: **Correct** | Last submitted: **10/10** | Times solved: **0** | Times unsolved: **2** | Score: **Medium** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Functions** | SubTopic: **None**

Results history: **Copy**

Show testcases scores Show solution

Student: kamalraj | Email id: 248701225@pj.dakshin.edu.in Test: REC_Python_Week 4_C00_Updated Course: NeoCodez_REC_C510021_Python Programming

 IP Address: 2405:fb99:102:100:ad73:b6ff:fe Test Details Test Resource: [Resource](#) Test Description: [Test Description](#) Diff User: [Diff User](#) Test Subject: [Test Subject](#) Recent User: [Recent User](#) Test Status: [Test Status](#)[Summary](#) [Details](#)[More](#) Coding (1)

Question No. 1

Single File Programming Question**Problem Statement:**

Imagine you are building a messaging application and you want to know the length of the messages sent by the users. You need to create a program that calculates the length of a message using the built-in function `len()`.

Input Format:

The input consists of a string representing the message.

Output Format:

The output prints an integer representing the length of the entered message.

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

The length of the string will be up to 100. The string includes lowercase, uppercase, numbers, space, and special characters.

Sample Test Cases:

Input:	Output:
hello	5

Input:	Output:
Today is a good day!	18

Input:	Output:
test	4

Analyze: Test

```
def your_code_here
    print(0)
    print(1)
    print(1)
    print(1)
```

Status: **Correct** | Last updated: **10/10** | Unlocked: **0** | Unlocked: **0** | Unlocked: **0** | Unlocked: **0** | **Score: 100%** | Question Type: **Single File Programming** | Subject: **Programming** | Topic: **Python** | Sub-Topic: **Fundamentals**

Show Test Case Scores Show Solution

Question No. 2

Single File Programming Question**Problem Statement:**

Implement a program that checks if a given number is Armstrong number or not.

Write a function `is_armstrong_number(number)` that checks if a given number is an Armstrong number or not.

Function Signature: `is_armstrong_number(number)`

Input Format:

The first line of the input consists of a single integer `n`, representing the number to be checked.

Sample Test Case:

The output should consist of a single line that displays a message indicating whether the input number is an Armstrong number or not.

Refer to the sample output for the formatting specifications.

Code constraints:

In the given test cases follow the following constraints:

1 ≤ n ≤ 10⁹.

Sample Test Cases :

Input 1:	Output 1:
153	153 is an Armstrong number.
Input 2:	Output 2:
123	123 is not an Armstrong number.

```
if __name__ == "__main__":
    num_to_check = int(input())
    sum_of_powers = 0
    temp = num_to_check
    while temp > 0:
        digit = temp % 10
        sum_of_powers += digit ** 3
        temp //= 10
    if sum_of_powers == num_to_check:
        print(f"{num_to_check} is an Armstrong number.")
    else:
        print(f"{num_to_check} is not an Armstrong number.)
```

Device: Desktop · Last submitted: 2020-07-16 · Status: Failed · 0 · Times attempted: 20 · Times submitted: 2 · Level: Medium · Question type: Single File Programming · Subject: Python · Tags: Functions · Code type: Python · Details

Brackets matching: Apply

Show testcases scores Show solution

Question No. 3

Single File Programming Question

Problem Statement

Imagine you are developing a text analysis tool for a cybersecurity company. Your tool needs to create a function that analyzes input strings to categorize and count the characters into four categories: uppercase letters, lowercase letters, digits, and special characters. The company wants this tool to process log files and identify potential security threats.

Function signature: analyze_string(input_string)

Input Format:

The input consists of a single string (without space), which may include uppercase letters, lowercase letters, digits, and special characters.

Output Format:

The first line contains an integer representing the count of uppercase letters in the format "Uppercase letters: [count]"

The second line contains an integer representing the count of lowercase letters in the format "Lowercase letters: [count]"

The third line contains an integer representing the count of digits in the format "Digits: [count]"

The fourth line contains an integer representing the count of special characters in the format "Special characters: [count]"

Refer to the sample output for the formatting specifications.

Code constraints:

In this scenario, the given test cases will not exceed the following constraints:

1) Length of the input string ≤ 100

Sample Test Cases :

Input 1:	Output 1:
Millennials	Uppercase letters: 1 Lowercase letters: 8 Digits: 0 Special characters: 8
Input 2:	Output 2:
123abc	Uppercase letters: 0 Lowercase letters: 3 Digits: 3 Special characters: 0

```
if __name__ == "__main__":
    analyze_string(input_string)
```

Header Output:

```
My code here
uppercase_count = 0
lowercase_count = 0
digit_count = 0
special_char_count = 0

for char in input_string:
    if 'A' <= char <= 'Z':
        uppercase_count += 1
    elif 'a' <= char <= 'z':
        lowercase_count += 1
    elif '0' <= char <= '9':
        digit_count += 1
    else:
        special_char_count += 1

return uppercase_count, lowercase_count, digit_count, special_char_count
```

Footer snippet

```
input_string = input()
uppercase_count, lowercase_count, digit_count, special_count = analyze_string(input_string)

print("Uppercase letters:", uppercase_count)
print("Lowercase letters:", lowercase_count)
print("Digits:", digit_count)
print("Special characters:", special_count)
```

Status: **Unlocked** | Last updated: **10/10** | Attempts: **0** | Times compiled: **0** | Times submitted: **1** | Level: **Medium** | Question type: **Single File Programming** | Subject: **Python** | Tools: **Condition** | Sub Topic: **None**

Block summary: **Apply**

Show testcases scores Show solution

Question No. 4

Single File Programming Question

Problem Statement:

This is a learning exercise about calculating powers. You want to implement a program that does the following:

1. calculate the result of raising a given base to a specific exponent using Python's built-in pow() function.
2. display intermediate powers from base⁰ to base^{exponent} as a list.
3. Calculate and display the sum of these intermediate powers.

Helpful hint: Write a helper function to calculate the intermediate powers.

Input Format:

The input consists of two space-separated two integer values representing base and exponent.

Output Format:

The first line prints the calculated result of raising the base to the exponent.

The second line prints a list of all powers from base⁰ to base^{exponent}.

The third line prints the sum of all those powers.

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

1. Base < 20
2. Exponent < 10

Sample test cases:

Input 1:	Output 1:
2 3	8 [1, 2, 4] 15

Input 2:	Output 2:
3 2	9 [1, 3] 10

```
My code here
def calculate_powers(base, exponent):
    result = base ** exponent
    print(result)
    intermediate_powers = []
    current_power = 1
    for i in range(1, exponent + 1):
        current_power *= base
        intermediate_powers.append(current_power)

    return intermediate_powers
```

```
def calculate_powers():
    base_of_powers = int(input("Enter base of powers"))
    exp = int(input("Enter exponent"))
    calculate_powers(base_of_powers, exp)
```

Submit Cancel Back to question 8/10 Home saved Draft unsaved From scratch Edit level Easy Question type Single File Programming Subject Programming Topic Python Full Topic Function

Show testcases scores Show solution

Question No. 5

Single File Programming Question

Problem Statement

You are developing a text-processing tool that checks if a given string starts with a specific character or substring. You need to implement a function that accepts a string and a character (or substring), and returns True if the string starts with the provided character (substring), or False otherwise.

Write a program that uses a lambda function to help find patterns in strings.

Input Format:

The first line contains a string 'str' representing the main string to be checked.

The second line contains a string 'n', which is the character or substring to check if the main string starts with it.

Output Format:

The last line of output gives 'True' if the string starts with the given character (substring), otherwise prints 'False'.

Refer to the sample for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1. less than 1000

Sample Test Cases :

Input 1:

```
Code editor  
x
```

Output 1:

False

Input 2:

```
Document viewer  
y
```

Output 2:

True

Input 3:

```
English  
ing
```

Output 3:

False

Input 4:

```
Java tutorial  
Java
```

Output 4:

True

Code Size : 1004 KB

Analyze

Test Run code

Wanna code here

```
main_string = input()
prefix = input()

starts_with = lambda s, p: s.startswith(p)
if starts_with(main_string, prefix):
    print('True')
else:
    print('False')
```

Status: **Solved** | More solutions: **1000** | Submissions: **0** | Time completed: **1** | Was submitted: **1** | Level: **Easy** | Question type: Single File Programming | Subject: Python | Tags: Functional Programming

See Topic

Last updated:

2 hours ago by

Apply

Show testcases scores Show solution

Student: kamoltrj | Email: 248701225@pj.duke.edu.in Test: R00_Python_Week 4_MQ Course: NewColab_R00_002321_Python Programming

Q Address: 248701225@pj.duke.edu.in Test: R00_Python_Week 4_MQ

Q Test Subject: — Test Result Date: May 6, 2020 (00:46 PM)

Q [0] User: 192.168.1.11 Test Subject Date: May 6, 2020 (00:46 PM)

Q Increase User: Chrome

Summary Details

☰ More

1 out of 25

Question No: 11

Multi Choice Type Question

What will be the output of the following code snippet?

```
1 def f(x,y,z):
2     return x + y + z
3 result = f(1, 1, 0)
4 print(result)
```

 A B C D

Status: Correct MarkedTotal: 0/1 HintsUsed: 0 Level: Easy QuestionType: MCQ-Multi-Select Subject: Python Topic: Functions TestTopic: Functions Bloom's Taxonomy: Understand

Question No: 12

Multi Choice Type Question

What will be the output of the following Python code?

```
1 def func(a,b,c=20):
2     print('a is ', a, ' b is ', b, ' AND c is ', c)
3 func(5, 7)
4 func(25, c = 34)
5 func(c = 98, a = 200)
```

 A None of the mentioned options B a is 5 and b is 7 and c is 10
a is 5 and b is 20 and c is 24
a is 100 and b is 100 and c is 5 C a is 2 and b is 5 and c is 10
a is 25 and b is 5 and c is 24
a is 5 and b is 5 and c is 30 D a is 2 and b is 10 and c is 10
a is 25 and b is 5 and c is 24
a is 5 and b is 5 and c is 100

Status: Wrong MarkedTotal: 0/1 HintsUsed: 0 Level: Easy QuestionType: MCQ-Single-Select Subject: Programming TestTopic: Functions Bloom's Taxonomy: Understand

 Show Solution

Question No: 13

Multi Choice Type Question

What will be the output of the following Python code?

```
1 def GCF(l1):
2     return l1[0] * l1[1]
3 print(GCF([10]))
4 print(GCF([10]))
```

 A 100 B 10 C 1000 D None of the mentioned options

Status: Correct MarkedTotal: 0/1 HintsUsed: 0 Level: Easy QuestionType: MCQ-Single-Select Subject: Programming TestTopic: Functions Bloom's Taxonomy: Understand

Show solution

Question No. 14

MULTI Choice Type Question

What is the best advantage of using lambda functions in Python?

 None of the mentioned options They can be used to define more complex functions than regular functions They allow you to write short code than regular functions They can be called directly from the command line without defining them. Status: Correct | Mark obtained: 0/1 | Revision level: 0 | Question type: MCQ Single Correct | Subject: Python | Topic: Functions | Sub-topic: Lambda function | Created by: anonymous | Analyse Show solution

Question No. 15

MULTI Choice Type Question

What is the output of the code shown?

```
1 def f(x):
2     global x
3     x+=1
4     print(x)
5 x=3
6 print(x)
```

→

 →

→

 Correctly answered Status: Wrong | Mark obtained: 0/1 | Revision level: 0 | Question type: MCQ Single Correct | Subject: Python | Topic: Variables | Sub-topic: Variables | Created by: anonymous | Understand Show solution

Student: kamalraj Email: 24870225@pj.du.ac.in Test: R00_Python_Week 4_MQ Course: NewColab_R00_002321_Python Programming

 IP Address: 2405:644:494:4179:240:89ff:fe:90d0
 TestDuration: 00:00:00:00:00:00 TestDuration: 00:00:00:00:00:00
 TestDuration: 00:00:00:00:00:00 Date: 2023-05-06
 Date: 2023-05-06 00:00:00 Browser Used: Chrome

Summary Details

☰ More

 0/4 (0%)

Question No: 1

Multi Choice Type Question

What will be the output of the following Python code?

```
1 def is_even(number):
2     if number % 2 == 0:
3         return True
4     result = is_even()
5     print(result)
```

A

False

 True

None of the mentioned options

Status: Correct Mark obtained: 10 Revision: 0 Level: Medium Question type: MCQ Single Correct Subject: Python Topic: Functions Sub-Topic: Lambda Functions Status: Incomplete Understood

 Show solution

Question No: 2

Multi Choice Type Question

What will be the output of the following Python code?

```
1 multiply = lambda x, y: x * y
2 print(multiply(2, "Hello"))
```

None of the mentioned options

 None of the mentioned options

True/False

None

Status: Correct Mark obtained: 10 Revision: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Functions Sub-Topic: Lambda Functions Status: Incomplete Understood

 Show solution

Question No: 3

Multi Choice Type Question

What will be the output of the following Python code?

```
1 def cube(x):
2     return x * x * x
3 x = cube(3)
4 print(x)
```

0

1

3

 27

Status: Correct Mark obtained: 10 Revision: 0 Level: Easy Question type: MCQ Single Correct Subject: Programming Topic: Python Sub-Topic: Functions Status: Incomplete Understood

 Show solution

Question No: 4

Multi Choice Type Question

What will be the output of the following code?

```
1 result = 10
2 result = -10
3 result = abs(result) + abs(result)
4 print(result)
```

D. 0

A. -20

B. 0

C. 20

Status: Correct | Mark obtained: 10 | Attempted: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: Functions | Sub-topic: Functions | Bookmarks: Underlined

Show solution

Question No. 5

Multi Choice Type Question

What will be the output of the following code?

```
1 value = 42
2 result = abs(value) + len(str(value))
3 print(result)
```

A. 43

B. 44

C. 42

D. 44

Status: Correct | Mark obtained: 10 | Attempted: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: Functions | Sub-topic: Functions | Bookmarks: Underlined

Show solution

Question No. 6

Multi Choice Type Question

What is the output of the code shown below?

```
1 def func():
2     x += 1
3     print(x)
4
5 global_variable = 10
6 func(global_variable)
7 print("Hello")
```

Compile and run

D. x

A. Hello

Status: Wrong | Mark obtained: 0 | Attempted: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: Functions | Sub-topic: Functions | Bookmarks: Underlined

Show solution

Question No. 7

Multi Choice Type Question

What will be the output of the following Python code?

```
1 def absolute_value(x):
2     if x < 0:
3         return -x
4     return x
5
6 result = absolute_value(-5)
7 print(result, absolute_value(5))
```

A. 0 0

B. 5 5

D. 5 -5

C. -5 5

Status: Correct | Mark obtained: 10 | Attempted: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: Functions | Sub-topic: Functions | Bookmarks: Underlined

Show solution

Question No: 8

Multi Choice Type Question

What will be the output of the following code?

```
1: x = -5
2: result = abs(x)
3: print(result)
```

 A B C D E Show solution

Question No: 9

Multi Choice Type Question

What will be the output of the following Python code?

```
1: def is_equal(x, y):
2:     if x == y:
3:         return x
4:     elif x == y:
5:         return "The numbers are equal"
6:     else:
7:         return y
8: print(is_equal(1, 1))
```

 A B C D E Show solution

Question No: 10

Multi Choice Type Question

What is the output of the following code snippet?

```
1: def my_function(a):
2:     a += 3
3:     return a
4:
5: a = 10
6: result = my_function(a)
7: print(a, result)
```

 A B C D E Show solution

Student: kamalraj | Email id: 248701225@projectkhan.edu.in | Test: R00_Python_Week 3_CW | Course: NeoColab_R60_OG23231_Python Programming

① Pending: #329550:24082040501219444454 | ② Timed Out: —
③ Testcases: 100/100④ Dif User: 100% | ⑤ Testcases: 100/100 | ⑥ Last Subm: Apr 8, 2023 10:00 AM
⑦ Testcases: 100/100 | ⑧ Last Subm: Apr 8, 2023 10:00 AM⑨ Recent User: Pratik
⑩ Testcases: 100/100[Summary](#) [Details](#)[More](#)

⑪ Coding (0)

Question No: 1

Single File Programming Question**Problem Statement:**

Write a program to check if a given string is perfect.

A perfect string must satisfy the following conditions:

- The string starts with a consonant.
- The string contains between consonants and vowels.
- Each centered character is a vowel.
- Vowels can occur consecutively multiple times but should not be followed immediately by a consonant.

The string satisfies all these conditions print "True", otherwise print "False".

Input Format:

The input consists of a string.

Output Format:

The output prints "True" if the string is perfect, otherwise print "False".

Refer to the sample output for formatting specifications.

Code constraints :

1. Length of the string <= 1000.

2. Strings must be in lowercase.

Sample Test Cases :

Input 1: aae	Output 1: True
Input 2: aaaa	Output 2: False
Input 3: aaaaa	Output 3: True
Input 4: aabb	Output 4: False
Input 5: aabbccdd	Output 5: False
Input 6: aaabbbcc	Output 6: False

Verify your code here

```

def is_perfect_string(s):
    vowels = set("aeiou")
    result = True
    if len(s) < 2:
        result = False
    else:
        consonants_left = len(s) - 1
        for i in range(len(s)):
            if i == 0:
                if s[i] in vowels:
                    result = False
                elif s[i] in consonants_left:
                    consonants_left -= 1
            else:
                if s[i] in vowels:
                    result = False
                elif s[i] in consonants_left:
                    consonants_left -= 1
                else:
                    if s[i-1] in vowels:
                        result = False
        return result
    return result
s = input("Enter String")
print("True" if is_perfect_string(s) else "False")

```

Python 3.8

Show history

Show testcase scores Show solution

Question No. 3

Single File Programming Question

Problem Statement:

Sophie is a technician who is responsible for maintaining web important documents. Both documents contain a certain placeholder placeholder that needs to be replaced with another character because they can be hacked. To ensure consistency in formatting, Sophie wants you to help her write a program that processes both documents by replacing the placeholder character with the new one.

Sophie also prefers a neat and structured output, so she wants you to ensure that both modified documents are printed in a single line, separated by a space, using the `format()` function.

Example:

Input:

Word

g

s

Output:

replaces

Explanation:

now in the character 'g' is replaced with 's' in the concatenated string.

Input Format:

The first line contains string1, the first document.

The second line contains string2, the second document.

The third line contains char1, the placeholder character that needs to be replaced.

The fourth line contains char2, the new character that will replace the placeholder.

Output Format:

The output displays in single line containing the modified string1 and string2, separated by a space.

Refer to the sample output for the formatting specifications.

Code constraints:

No given test cases fall under the following conditions:

The strings may contain any primitive characters.

char1 and char2 are always single characters.

The maximum length of each string is 100 characters.

Sample Test Cases :

Input 1:

Hello

World

o

s

Output 1:

Hello World

Input 2:

0000

1111

2

Output 2:

0000 1111

Input 3:

Python programming

Practices

3

Output 3:

Python Programming Practices

Python code here:

```
def1 = input("String1")
def2 = input("String2")
char1 = input("char1")
char2 = input("char2")
modified_def1 = def1.replace(char1, char2)
modified_def2 = def2.replace(char1, char2)
print("{} {}".format(modified_def1, modified_def2))
```

Python 3.6

status: correct | last updated: **May 10, 2023** | time taken: 0 | time compiled: 1 | time submitted: 2 | level: Medium | question type: Single File Programming | major: Python | topic: Datatype | submitter: steeg

Show history

Show testcases scores Show solution

Single File Programming Question**Problem Statement:**

A company is creating email accounts for its new employees. They want to use a naming convention for email addresses that consists of the first letter of the employee's first name, followed by their last name, followed by @company.com.

The company also has a corporate email domain for administration employees.

Write a program that prompts the user for their first name, last name, role, and company and then generates their email address using the appropriate naming convention based on their role. This is demonstrated in the below example.

Note:

The generated email address should consist of the first letter of the first name, the last name in lowercase, and @suffix based on the company, and @domain.

Input Format:

The first line of input consists of the first name of an employee (as a string).

The second line consists of the last name of an employee (as a string).

The third line consists of the role of the employee (as a string).

The fourth line consists of the company name (as a string).

Output Format:

The output consists of a single line containing the generated email address for the employee, following the specified naming convention.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

All input strings [first_name, last_name, role, company] consist of only lowercase and uppercase English letters.

Sample Test Cases:**Input 1:**

```
John
Smith
Admin
Salesforce
```

Output 1:

```
jsmith@salesforce.com
```

Input 2:

```
John
Smith
IT
Salesforce
```

Output 2:

```
jsmith@it_salesforce.com
```

If you code here

```
first_name = input("First Name")
last_name = input("Last Name")
role = input("Role")
company = input("Company")
username = first_name[0].lower() + last_name.lower()
if role.lower() == "admin":
    email = f"{username}@{company}.com"
else:
    email = f"{username}@{company.lower()}.com"
print(email)
```

Status: **Correct** | Mark Allocated: **10/10** | Last updated: **0** | Times compiled: **4** | Times submitted: **2** | Score: **100** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Data Type** | Sub Topic: **String**

Show test case scores Show solution

Single File Programming Question

Problem Statement

You are developing a tool for a cybersecurity company. Your task is to analyze input strings to categorize and count the uppercase, lowercase, uppercase letters, lowercase letters, digits, and special characters. The company needs this tool to process log files and identify potential security threats.

Input Format:

The input consists of a single line containing the string A .

Output Format:

The output consists of four lines:

The first line contains an integer representing the count of uppercase letters in the format "Uppercase letters: [uppercase count]".

The second line contains an integer representing the count of lowercase letters in the format "Lowercase letters: [lowercase count]".

The third line contains an integer representing the count of digits in the format "Digits: [digit count]".

The fourth line contains an integer representing the count of special characters in the format "Special characters: [special character count]".

Refer to the sample output for the formatting specifications.

Code Constraints:

In this exercise, the given test cases will fall under the following constraints:

Input string may contain a mix of characters, including alphanumeric, symbols, and special characters.

Length of string A is 100.

Sample Test Cases:

Input 1:

```
mississippi
```

Output 1:

```
Uppercase letters: 1  
Lowercase letters: 6  
Digits: 0  
Special characters: 8
```

Input 2:

```
learning
```

Output 2:

```
Uppercase letters: 0  
Lowercase letters: 6  
Digits: 0  
Special characters: 2
```

Please code here

```
def analyze_log_entry(log_entry):  
    log_entry = log_entry()  
    uppercase_count = 0  
    lowercase_count = 0  
    digit_count = 0  
    special_count = 0  
    for char in log_entry:  
        if 'A' <= char <= 'Z':  
            uppercase_count += 1  
        elif 'a' <= char <= 'z':  
            lowercase_count += 1  
        elif '0' <= char <= '9':  
            digit_count += 1  
        else:  
            special_count += 1  
    print(f"Uppercase letters: {uppercase_count}")  
    print(f"Lowercase letters: {lowercase_count}")  
    print(f"Digits: {digit_count}")  
    print(f"Special characters: {special_count}")  
    return f"Log entry: {log_entry}"  
analyze_log_entry(log_entry)
```

Python 3.8

Home Logout Mark Unread Kudo Unlocked 0 Times compiled 1 Times submitted 1 Level: Medium Question type: Single File Programming Subject: Programming Topic: Python Sub Topic: Range

Show testcases scores Show solution

Question No. 3

Single File Programming Question

Problem Statement

Write a program using operations in python environment to practice using built-in functions. The input given is string A , and the task is to:

- Find the length of the string using str len function.
- Copy the content of this string into another string B using built-in functionality.

Help: You can write a program that efficiently performs three operations:

Input Format:

The input consists of a single line containing the string A (without spaces).

Output Format:

The first line of output prints the length of the given string.

The second line prints the copied string without any extra spaces at the end.

Refer to the sample output for the formatting specifications.

Code Constraints:

The given test cases fall under the following constraints:

0 < len of the string A < 100.

Sample Test Cases:

Input:

technology

Output 1:

Length of the string: 11
Output string: technology

Input 2:

maths-is-easy

Output 2:

Length of the string: 11
Output string: maths-is-easy

Write your code here

System.out.println();

```

public string string_operations_forswitch()  

{
    string s = input();
    length s = length(string s);
    string t = string s;
    int l = length of the string; length s();
    for (int i = 0; i < l; i++)
        if (string s[i] != s[i + 1])
            t += s[i];
    return t;
}
string string_operations_forswitch();

```

Status: **Solved** | More problems | **ANSWER** | Unlocked: 0 | Times compiled: 1 | Times submitted: 1 | **level**: Medium | Question type: Single File Programming | Subject: X | Topic: Strings | Tags: Basic, Strings

Show test case scores Show solutions

Question no: 6

Single File Programming Question

Problem statement:

Accept an unsorted list of length n with both positive and negative integers, including 0. The task is to find the smallest positive number missing from the array. Assume the n value is always greater than zero.

Input format:

The first line consists of n, which means the number of elements in the array.
The second line consists of the values in the list as space-separated integers.

Output format:

The output displays the smallest positive number, which is missing from the array.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:
1 ≤ n ≤ 100.

Sample test cases :

Input:

5

Input 2:

1 2 3 5 4

Input 3:

1 2 2 3 4

Output 1:

1

Output 2:

3

Output 3:

5

Write your code here

System.out.println();

```

public int find_smallest_missing_positive()  

{
    int n = Integer.parseInt(input());
    String str = input().split(" ");
    int arr[] = new int[n];
    int present_positions = 0;
    for (int i = 0; i < n; i++)
        if (arr[i] == 0)
            present_positions++;
    int smallest_missing = 1;
    while (smallest_missing == present_positions)
        smallest_missing++;
    return smallest_missing;
}
int find_smallest_missing_positive();

```

Show test case scores Show solution

Question No. 5

Single File Programming Question

Problem Statement

Dawn was doing her homework. She needed to write a paragraph about Indian history. During that time, she noticed that some words were repeated frequently. She wanted counting the number of times a particular word was repeated.

Your task is to Help Dawn to write a program to get a string from the user. Count the number of times a word is repeated in the string.

Refer. Coderunner.py

Input Format:

The first line of input contains a string str1.

The second line contains a single word str2 which needs to be counted, str2.

Output Format:

The output displays the number of times the given word occurs in the string.

If the second string str2 is not present in the first string str1, prints 0.

Refer to the sample output for the formatting specifications.

Code constraints

The given test cases fall under the following conditions:

The input string may contain any printable characters, including spaces, punctuation marks, and alphanumeric characters.

Sample Test Cases :

Input 1:
I like happy because I saw the others were happy and because I know I should feel happy.
happy

Output 1:
3

Input 2:
If you talk the truth, you don't have to remember anything, a lie can travel half way around the world while a
truth

Output 2:
0

Input 3:
The sun
sun

Output 3:
0

Input 4:
Litter on a slide, with your
slide

Output 4:
1

Run your code here

```
from string import punctuation
str1=input()
str2=input()
for p in punctuation:
    str1=str1.replace(p,'')
    str2=str2.replace(p,'')

words1=str1.split()
count_words1=len(words1)
words2=str2.split()
count_words2=len(words2)

for i in range(count_words1):
    count=0
    for j in range(count_words2):
        if words1[i]==words2[j]:
            count+=1
    print(count)
```

Python 3

Show test case scores Show solution

Single File Programming Question**Problem Statement:**

You are analysing a series of measurements taken over time. You need to identify all the 'peaks' in the set of measures.

A peak is defined as an element that is greater than its immediate (neighbors) boundary elements are considered peaks if they are greater than their neighbors.

Your task is to find the list of such peaks using list comprehension.

Example:**Input:**

1 2 3 4 5 6 7 8 9 10 11

Output:

Peaks: [3, 4, 7, 10, 11]

Explanation:

3 is a peak because it's greater than 1 and 2.

4 is a peak because it's greater than 2 and 5.

7 is a peak because it's greater than 5 and 8.

10 is a peak because it's greater than 8 and 11.

8 is a peak because it's an boundary element and it's greater than 7.

Input Format:

The input consists of several integers separated by spaces, representing the measurements.

Output Format:

The output displays 'Peaks:' followed by a list of integers, representing the peak elements in the list.

Refer to the sample output for the formatting specifications.

Code constraints:

1 ≤ Total elements ≤ 25

Sample Test Cases:**Input 1:**

1 2 3 4 5 6 7 8 9 10 11

Output 1:

Peaks: [3, 4, 7, 10, 11]

Input 2:

1 2 3

Output 2:

Peaks: [1]

Your code here

```
arr = [int(x) for x in input().split()]
peaks = [arr[i] for i in range(len(arr))
         if ((arr[i] > arr[i - 1]) and (arr[i] > arr[i + 1])) or
            ((arr[i] < arr[i - 1]) and (arr[i] < arr[i + 1])) or
            ((arr[i] > arr[i - 1] and arr[i] > arr[i + 1]) and (arr[i - 1] < arr[i] < arr[i + 1]))]
```

Python 3.7

Status: **Solved** | Last submitted: **10/10** | Run count: **0** | Time completed: **8** | Errors submitted: **2** | Level: **Hard** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Data Structures in python**

Show testcases scores Show solution

Student: kamalraj | Email id: 248701225@projdatashm.edu.in Test: R00_Python_Week3_C00 Course: RedCeBb_R00_OG3221_Python Programming

 Pending: 48.30%
 Incomplete: 51.60% Timed Out: —
 Last Test Case: April 06, 2023 10:00 AM DFL Test: Pending
 Last Submited: April 06, 2023 10:00 AM Review Used: Pending
 Last Submited: April 06, 2023 10:00 AM[Summary](#) [Details](#)[More](#)[Coding \(1\)](#)

Question No: 1

Single File Programming Question**Problem Statement:**

You are working on a Python program to manage a list of elements. You need to append multiple elements to the list and then remove an element from the list at a specified index.

Your task is to create a program that takes file managing the list. The program should allow user to input a list of elements, append them to the existing list, and then remove an element at a specified index.

Input Format:

The first line contains an integer n representing the number of elements to be appended to the list.

The next n lines contain integers, representing the elements to be appended to the list.

The final line of input contains an integer M , representing the index of the element to be popped from the list.

Output Format:

The first line of output displays the original list.

The second line of output displays the list after popping the element at the index M .

The third line of output displays the popped element.

Refer to the sample output for the formating specifications.

Code constraints:

In this scenario, the given test cases will fall under the following constraints:

$1 \leq n \leq 20$

$0 \leq M \leq n$

Sample test cases:**Input 1:**

```
4
24
36
1
5
-1
```

Output 1:

```
List after appending elements: [24, 36, -1, 5]
List after popping last element: [24, 36, -1, 10]
Popped element: 5
```

Input 2:

```
4
24
36
10
5
-1
```

Output 2:

```
List after appending elements: [24, 36, 10, 5]
List after popping last element: [24, 36, 5]
Popped element: 10
```

My code here:

```
def append():
    n = int(input())
    elements = []
    for i in range(n):
        element = int(input())
        elements.append(element)
    print("List after appending elements: ", elements)
    print("List after popping last element: ", elements.pop())
    print("Popped element: ", elements.pop())
    print("List after popping last element: ", elements)
    print("Popped element: ", elements.pop())
append()
```

Python 3.9

Status: Correct Mark placement: Auto-grade: Times compiled: 8 Times submitted: 1 Reset Easy Checked type: Single File Programming Subject: Python Topic: DataTypes Submission Date: 2023-04-06 10:00:00

Show test case scores Show All Submissions

Question No. 2

Single File Programming Question

Problem Statement

You have a string consisting of a phone number in the format '(100) 100-1000'. You need to extract the area code from the phone number and create a new string that contains only the area code.

Write a Python program for the same.

Note

(100) - Area code
100-1000 - Phone number

Input Format:

The input consists of a string representing the phone number in the format '(100) 100-1000'.

Output Format:

The output displays "Area code:" followed by a string representing the area code for the given phone number.

Refer to the example output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

The input phone number consists of eight units.

Sample Test Cases:

Input:

(123) 456-7890

Output 1:

Area code: 123

Input 2:

(123) 111-1111

Output 2:

Area code: 123

Edit your code here

```
print("Print\n")
def extract_area_code():
    phone_number = input()
    match = re.search(r'(\d{3})\d{3}\d{4}', phone_number)
    if match:
        area_code = match.group("area_code")
        print("Area code: " + area_code)
    else:
        print("Invalid phone number format.")

if __name__ == "__main__":
    extract_area_code()
```

Python 3.8

Status: **Blocked** | Marked solved: **NO** | Attempts: **0** | Time compiled: **0** | Time submitted: **0** | Level: **Easy** | Question type: **Single File Programming** | Subject: **Python** | Tags: **Descriptive** | Out Topic: **String**

Show testcases scores Show solution

Question No. 3

Single File Programming Question

Problem Statement

Given a list of positive and negative numbers, arrange them such that all negative integers appear before all the positive integers in the array. The order of appearance should be maintained.

Example:

Input:
[2, 3, -1, -2, 4, -3, 5, -1, -6]

Output:
[-6, -3, -2, -1, 2, 3, 4, 5]

Explanation:
The output is the rearranged list where all the negative integers appear before the positive integers while maintaining the original order of appearance.

Input Format:

The input consists of a single line containing a list of integers enclosed in square brackets separated by commas.

Output Format:

The output displays "[list]" followed by an arranged list of integers as required, separated by commas and enclosed in square brackets.

Refer to the example output for the formatting specifications.

Code constraints:

In this question, the given test cases will fall under the following constraints:

-1000 ≤ array elements ≤ 1000

Sample Test Cases:

Input:

[2, 3, -1, -2, 4, -3, 5, -1, -6]

Output 1:

[-6, -3, -2, -1, 2, 3, 4, 5]

[11, 12, -13, 14, 15, 16, -17, -18]

ans = [-11, -12, 13, 14, 15, 16, -17, -18]

Input 2:

[11, 12, -13, 14, 15, 16, -17, -18]

Output 2:

ans = [-11, -12, 13, 14, 15, 16, -17, -18]

My code here

```
#!/usr/bin/python3
#Sort a list of integers, maintaining them as negative numbers
#case before positive numbers while maintaining their original order,
#and prints the numbers like with the specified format.
#-----
#Import time > input()
numbers_str = input().split(',')
numbers = [int(num) for num in numbers_str]
negative_numbers = [num for num in numbers if num < 0]
positive_numbers = [num for num in numbers if num >= 0]
rearranged_list = negative_numbers + positive_numbers
print(''.join(str(i) for i in rearranged_list))

#-----
```

reverse_numbers_sorted()

Status | Current | Last submitted | 10/10 | Run successful | 2 | Times submitted | 1 | level: Medium | Question type: Single File Programming | Submit Python | Edit | Delete | Edit Test | See

WorldRankings Reply

Show testcases scores Show solution

Question No. 4

Single File Programming Question

Problem Statement

Create a simple program to slice a given string based on user-defined start and end positions.

The program should check whether the provided position is valid and then return the sliced portion of the string if the positions are within the string's length.

Input Format:

The first line contains the input string as a string.
The second line contains the start position (0-based index) as an integer.
The third line contains the end position (0-based index) as an integer.

Output Format:

The output displays the following format:
If the start and end positions are valid, print the sliced string.
If the start and end positions are invalid, print "Invalid start and end positions".

Refer to the sample output for formatting specifications.

Code constraints :

The input string consists of printable characters and may contain letters, numbers, special symbols, and spaces.
1 <= The input string length <= 100
0 <= start_pos <= end_pos < len(input_string)

Sample test cases :

Input 1:

pythontutorial

Output 1:

python

Input 2:

abcde

Output 2:

abc

Input 3:

model@192.168.1.100

Output 3:

Invalid start and end positions

My code here

```
#!/usr/bin/python3
input_string = input()
start_pos = int(input())
end_pos = int(input())

string_length = len(input_string)

if 0 <= start_pos <= end_pos < string_length:
    sliced_string = input_string[start_pos : end_pos + 1]
    print(sliced_string)
else:
    print("Invalid start and end positions")
```

slice_string_rearranged()

```
# reverse_concatenate()
```

Status: Pending review Last reviewed: 18/03 Reviewer: Times compiled: 0 Times submitted: 4 User: Deep Question type: SingleFile Programming Subject: Python Team: DataTypes Sub Team: String
Review history: Apply

Show testcases scores Show solution

Question No. 5

Single File Programming Question

Problem Statement:

Armin is working on a program to manipulate strings. He wants to create a program that takes two strings as input, reverses the second string, and then concatenates it with the first string.

Armin needs your help to design a program.

Input Format:

The input consists of two strings, in separate lines.

Output Format:

The output displays a single line containing the concatenation of the first string and the reversed second string.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fall under the following constraints:

Use only '+' operator to concatenate.

Do not use join or format.

The length of strings are at most 100 characters each without spaces.

Sample test cases:

Input:

```
India  
world
```

Output 1:

```
IndiaWorld
```

Input 2:

```
apple  
banana
```

Output 2:

```
appleanaban
```

Input 3:

```
the_sun  
moon
```

Output 3:

```
the_sunmoon
```

Write your code here:

```
# reverse_concatenate()  
def reverse_concatenate():  
    string1 = input()  
    string2 = input()  
  
    reversed_string2 = string2[::-1]  
    result = string1 + reversed_string2  
    print(result)  
  
reverse_concatenate()
```

Status: Pending Last reviewed: 18/03 Reviewer: Times compiled: 0 Times submitted: 1 User: Deep Question type: SingleFile Programming Subject: Python Team: DataTypes Sub Team: String
Review history: Apply

Show testcases scores Show solution

Student: kamalraj | Email id: 248701225@pj.dakshin.edu.in Test: R00_Python_Week 3_MQ Course: NeoClass_R00_052320_Python Programming

 IP Address: 2405:4844:4:400:200:1000:100:1000
 Test resource: 60000 Test duration: —
 Test start time: Apr 10, 2023 11:33:04 AM Device Used: Windows
 Test history since: Apr 10, 2023 11:33:04 AM Device Used: Chrome
 Test history since: Apr 10, 2023 11:33:04 AM

Summary Details

More

 [Multi-choice](#)

Question No: 21

Multi Choice Type Question

What is the output of the following Python code?

```
[1] txt = "My Classmate"
[2] print(txt, end="")
[3] print(txt, end="\n")
```

 A B C D

CORRECT ANSWER

Status: correct Marked correct Unmarked: 0 Level: Easy Question type: MCQ Single correct Subject: Python Topic: Data types Sub topic: String Recomendation: Remember Show solution

Question No: 22

Multi Choice Type Question

What is the result of the slicing operation m[-3:-1] on the list m = [1, 2, 3, 4, 5, 6]?

 A B CStatus: correct Marked correct Unmarked: 0 Level: Easy Question type: MCQ Single correct Subject: Python Topic: Data structures Sub topic: List Recomendation: Remember Show solution

Question No: 23

Multi Choice Type Question

Which method is used to add element from the end of a list?

 A B CStatus: correct Marked correct Unmarked: 0 Level: Easy Question type: MCQ Single correct Subject: Python Topic: Data Structures in python Sub topic: List comprehension in python Recomendation: Remember Show solution

Question No: 24

Multi Choice Type Question

What does negative indexing in Python lists allow you to do?

 A B C

Access elements in the list forward

Access elements in the list from the start

Sort elements in the list

Modify elements in the list

Status: Correct Check solution Wrong Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data Structures Sub Topic: List Shows answerkey Remember

Show solution

Question No: 25

MULTI Choice Type Question

Suppose list is [2, 33, 111, 14, 22]. What is len([::])?

Err

0

[2, 33, 111, 14]

[22, 14, 111, 33, 2]

Status: Wrong Check solution Correct Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data Structures Sub Topic: List Shows answerkey Remember

Show solution



Student: kamalraj | Email id: 248701225@pj.dakshin.edu.in Test: R00_Python_Week3_MCQ Course: NeoClass_R00_052320_Python Programming

 IP Address: 2405:4844:4:4:6:0:0:0:0:0:0:0:0:0:0:0 Test Details Date Used: 19/04/2023 Browser Used: Chrome Test Resource Test Start Time: Apr 19, 2023 11:33:04 AM Test End Time: Apr 19, 2023 11:33:41 AM Test Duration: 1 min Test Status Test [Multi Choice](#)

Question No: 11

Multi Choice Type Question

What is the output of the following Python code?

1. name = "John"
 2. age = 21
 3. message = "My name is %s and I am %s years old." % (name, age)
 4. print(message)

My name is John and I am 21 years old.

My name is Tom and I am 21 years old.

 My name is John and I am 21 years old.

My name is Tom and I am 21 years old.

 status: Correct | mark: 0/1 | submitted: 0 | level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: DataType | Sub-Topic: String | Previous Question | Next Question | Understand Show Solution

Question No: 12

Multi Choice Type Question

What is the output of the following Python code?

1. word = "Python"
 2. result = word[-1]
 3. print(result)

 Python Python Python status: Wrong | mark: 0/1 | submitted: 0 | level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: DataType | Sub-Topic: String | Previous Question | Next Question | Understand Show Solution

Question No: 13

Multi Choice Type Question

What is the output of the following Python code?

1. word = "Python"
 2. result = word[-1:-1]

 Python nothing Python nothing status: Wrong | mark: 0/1 | submitted: 0 | level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: DataType | Sub-Topic: String | Previous Question | Next Question | Understand Show Solution

Question No: 14

Multi Choice Type Question

What is the output of the following Python code?

```
[1] string1 = "Hello"
[2] string2 = "World"
[3] result = string1 + string2
[4] print(result)
```

Hello World

HelloWorld

WorlHello

World Hello

Status: Correct Mark obtained: 100 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: DataTypes Sub Topic: String Home > Home > DataTypes > String > Home's answer > Understand

Show solution

Question No: 15

Multi Choice Type Question

Which method in Python is used to create an empty list?

empty_list()

[]

create_list()

Status: Correct Mark obtained: 100 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: DataTypes Sub Topic: List Home > Home's answer > Understand

Show solution

Question No: 16

Multi Choice Type Question

What is the output of the following Python code?

```
[1] word = "programmer"
[2] answer = word[-index("er")]
[3] print(answer)
```

e

r

m

Status: Correct Mark obtained: 100 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: DataTypes Sub Topic: List Home > Home's answer > Understand

Show solution

Question No: 17

Multi Choice Type Question

What does the following code output?

```
[1] list = [10, 20, 30, 40, 50]
[2] print(list[-4:-1])
```

[10, 40]

[10, 50]

[10, 20, 40]

Status: Correct Mark obtained: 100 Released: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data Structures Sub Topic: List Home > Home's answer > Understand

Show solution

Question No: 18

Multi Choice Type Question

What is the output of the following python code?

```
text = "Python"
result = text.center(16, "=")
print(result)
```

Python

Python!!!!

Python

Python!!!!

Status: Wrong | Marked correct: 0/1 | Unmarked: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: DataTypes | Sub Topic: String | Previous question | Next question | Remember

Show solution

Question No. 18

Multi Choice Type Question

Suppose with [2, 33, 222, 4, 25], what is sort()?

None

None

33

None of the mentioned options

Status: Correct | Marked correct: 0/1 | Unmarked: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: Data Structures | Sub Topic: List | Previous question | Next question | Remember

Show solution

Question No. 19

Multi Choice Type Question

What is the output of the following python code?

```
a = "Hello"
b = "World"
c = a + " " + b
print(c)
```

Console time box

Hello
World

Hello World

HelloWorld

Status: Correct | Marked correct: 0/1 | Unmarked: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Python | Topic: DataTypes | Sub Topic: String | Previous question | Next question | Remember

Show solution

Student: kamalraj | Email id: 248701225@pjmskhan.edu.in Test: R00_Python_Week 3_MQ Course: NeoClass_R00_052320_Python Programming

IP Address: 2405-004-446-0010/0000000000000000
Last accessed: 10-Apr-2023 11:33:04 AMDevice Used: Windows
Last visited: 10-Apr-2023 11:33:04 AMDevice Used: Chrome
Last visited: 10-Apr-2023 11:33:04 AM

Summary Details

More

 [Multi-choice](#)

Question No: 1

Multi Choice Type Question

Suppose list is [4, 3, 2, 1, 5]. Which of the following is the correct syntax for slicing operator?

- print(list[2])
- print(list[1])
- print(list[-1])

 [All of the mentioned options](#)Status: [Correct](#) Mark obtained: **10** / 10 Last accessed: **10** Level: **Easy** Question type: **MCQ Single correct** Subject: **Python** Topic: **DataTypes** Sub Topic: **list** Previous Question Remember Show solution

Question No: 2

Multi Choice Type Question

Which of the following is a valid way to use the '+' operator to concatenate strings in Python?

- string + string2
- "Hello" + string2
- string * string2
- string + "Hello"

 [Both A and C are correct](#)Status: [Correct](#) Mark obtained: **10** / 10 Last accessed: **10** Level: **Easy** Question type: **MCQ Single correct** Subject: **Python** Topic: **DataTypes** Sub Topic: **String** Previous Question Remember Show solution

Question No: 3

Multi Choice Type Question

What will be the output of the following code?

```
[1] numbers = [1, 2, 3, 4, 5]
[2] numbers.reverse()
[3] print(numbers)
```

- [1, 2, 3]
- [1, 3, 2]
- [1, 2, 3, 4, 5]

 [None of the mentioned](#)Status: [Correct](#) Mark obtained: **10** / 10 Last accessed: **10** Level: **Easy** Question type: **MCQ Single correct** Subject: **Python** Topic: **DataTypes** Sub Topic: **List** Previous Question Remember Show solution

Question No: 4

Multi Choice Type Question

What is the output of the following Python code?

```
[1] b = "Project"
[2] print(b[2:5])
```

Page 1

Q1 Single Line

1

D 100

status: correct | Marked correct: 0 | Retried: 0 | Level: Easy | Question type: MCQ-Single correct | Subject: Python | Topic: Data Type | Sub Topic: List | Recent activity: 1 hour ago | Understand

Show solution

Question No. 5

Multi Choice Type Question

What will be the output of the following program?

```
numbers = [1, 2, 3, 4, 5]
for i in numbers:
    print(i)
```

D 100

[1, 2, 3, 4, 5, 6, 7]

[1, 2, 3, 4, 5, 6, 7, 8]

Q Complete this code

status: Wrong | Marked correct: 0 | Retried: 0 | Level: Easy | Question type: MCQ-Single correct | Subject: Python | Topic: Data Type | Sub Topic: List | Recent activity: 1 hour ago | Understand

Show solution

Question No. 6

Multi Choice Type Question

What will be the output of the following code?

```
my_list = [1, 2, 3, 4]
print(my_list[2])
```

1

D 100

2

3

status: correct | Marked correct: 0 | Retried: 0 | Level: Easy | Question type: MCQ-Single correct | Subject: Python | Topic: Data Structure | Sub Topic: List | Recent activity: 1 hour ago | Understand

Show solution

Question No. 7

Multi Choice Type Question

If you have a list `list = [1, 2, 3, 4, 5, 6]`, what does the slicing operation `list[-2:]` result?

D The last three elements of the list

An index out of range error

The middle three elements of the list

The first three elements of the list

status: correct | Marked correct: 0 | Retried: 0 | Level: Easy | Question type: MCQ-Single correct | Subject: Python | Topic: Data Structure | Sub Topic: List | Recent activity: 1 hour ago | Understand

Show solution

Question No. 8

Multi Choice Type Question

What is the output of the following code?

```
my_list = [1, 2, 3]
my_list[4] = 1
print(my_list)
```

Q-9

Status: Correct Mark obtained: 10 Time taken: 0 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data Structures Sub Topic: List Score: 100% Status: Unanswered Understand

Show solution

Question No: 9

Multi Choice Type Question

What is the output of the following code?

```
my_list = [5, 6, 1, 2, 3, 4]
for item in my_list:
    print(item)
```

None

True

None of the mentioned options

Q-10

Status: Correct Mark obtained: 6 Time taken: 6 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data Structures Sub Topic: List Score: 100% Status: Unanswered Understand

Show solution

Question No: 10

Multi Choice Type Question

What does the append() method do in Python?

Join two lists together

Q-10 Append element to the end of the list

Decrease the last element of the list

Increase the order of the list

Status: Correct Mark obtained: 6 Time taken: 6 Level: Easy Question type: MCQ Single Correct Subject: Python Topic: Data Structures in python Sub Topic: List comprehension in python Score: 100% Status: Unanswered Remember

Show solution

Student: kamalraj | Email id: 248701225@projektaakhm.edu.in Test: R00_Python_Week2_PA1_Updated Course: NeoColab_R00_OG23231_Python Programming

IP Address: 240.20.199.122 MacAddress: b8:00:45:00:00:05

Tom-Delaney

File test1.txt (Tue, Mar 22, 2022, 11:33 AM)

Dhaval Patel

File test1.txt (Tue, Mar 22, 2022, 11:33 AM)

Bhavin Patel

File test1.txt (Tue, Mar 22, 2022, 11:33 AM)

[Summary](#) [Details](#)[More](#)[Coding \(W\)](#)

Question No: 1

Single File Programming Question**Problem Statement:**

Karnail recently started another business. His company wants to encourage the consumers to switch to pay-as-you-go usage. The electricity company charged different rates based on the number of units consumed. For the first 100 units, there is no charge. For units consumed beyond 100 and up to 200, there is a charge of Rs. 5 per unit. For units consumed beyond 200, there is a charge of Rs. 10 per unit.

Write a program to help Karnail calculate the amount, who needs to pay for his electricity bill based on the units consumed.

Input Format:

The input consists of an integer, representing the number of units.

Output Format:

The output prints the total amount of the electricity bill, an integer indicating the amount/kmata needs to pay in the format "Rs. amount".

Refer to the sample output for the exact format.

Code constraints:

The given test cases fall under the following constraints:

1 ≤ units ≤ 10000000

Sample Test Cases :

Input 1: 100	Output 1: Rs. 0
Input 2: 200	Output 2: Rs. 50
Input 3: 300	Output 3: Rs. 0

```
public class Main {
    public static void main(String[] args) {
        int units = Integer.parseInt(args[0]);
        if (units <= 100)
            System.out.println("Rs. 0");
        else if (units > 100 & units <= 200)
            System.out.println("Rs. " + ((units - 100) * 5));
        else
            System.out.println("Rs. " + ((units - 200) * 10));
    }
}
```

Status: **Accepted** | Monitored by: **NYC** | Submissions: 0 | Times compiled: 1 | Times submitted: 1 | **Medium** | Question type: Single File Programming | Student: Python | Topic: Control Structures

Bit Topic: Conditional Statement | Return statement | Apply

Show testcase scores Show solution

Question No: 2

Single File Programming Question**Problem Statement:**

A series is associated by the concept of summing numbers sequentially based on their properties. We plan to write a program that calculates the sum of even numbers and odd numbers sequentially from 1 to a given positive integer. Assume user to input an integer value to represent the upper limit of the range. Help user by developing a program that computes and displays the sum of even and odd numbers separately.

Input Format:

The input consists of a single integer N , where it is the upper limit of the range.

Output format:

The output consists of two lines:

The first line displays the sum of even numbers from 1 to N .

The second line displays the sum of odd numbers from 1 to N .

Refer to the sample output for the exact format.

Code constraints:

The given test cases fall under the following conditions:

$1 \leq N \leq 30$

Sample test cases :

Input 1:

10

Output 1:

Sum of even numbers from 1 to 10 is: 30
Sum of odd numbers from 1 to 10 is: 25.

Input 2:

5

Output 2:

Sum of even numbers from 1 to 5 is: 10
Sum of odd numbers from 1 to 5 is: 9.

Your code here

```
# You are writing Python code
N = int(input())
sum_even = 0
sum_odd = 0
for i in range(1,N+1):
    if i%2==0:
        sum_even+=i
    else:
        sum_odd+=i
print("Sum of even numbers from 1 to {} is: {}".format(N,sum_even))
print("Sum of odd numbers from 1 to {} is: {}".format(N,sum_odd))
```

Python 3.7.4

Status: Correct

Last submitted: 07/01/2019

Run status: 0

Time elapsed: 1s

Time submitted: 1s

code: 5

Topic: 1

Question type: Single File Programming

Subject: Python

Topic: Control Structures

Task Topic: Looping Statement

Source repository: Apply

Show Testcase scores Show solution

Question No. 3

Single File Programming Question

Input/Output Statement:

An instructor requires your guidance to develop a program that facilitates the identification of leap years in a specified range. Your task is to create a program that takes two integer inputs, representing the start and end points of the range, and then prints all the leap years within that range.

Input format:

The first line of the input consists of an integer, which represents the start year.

The second line consists of an integer, which represents the end year.

Output format:

The output displays the leap years within the given range, separated by newlines.

Refer to the sample output for formatting specifications.

Code constraints:

The given test cases fall under the following conditions:

$1000 \leq \text{start year} \leq \text{end year} \leq 3000$

Sample test cases :

Input 1:

1000
2000

Output 1:

2000
2004
2008
2012
2016
2020
2024
2028
2032
2036
2040
2044
2048
2052

Input 2:

Output 2:

2149
2150

2151
2152
2153
2154
2155
2156
2157
2158

Play your code here

```
#include <iostream>
using namespace std;
int fact(int year)
{
    int sum = 0;
    int digit;
    while (year != 0)
    {
        digit = year % 10;
        sum += digit * fact(digit);
        year /= 10;
    }
    return sum;
}
int main()
{
    int start, end;
    cout << "Enter start year : ";
    cin >> start;
    cout << "Enter end year : ";
    cin >> end;
    for (int year = start; year <= end; year++)
    {
        if (year == fact(year))
            cout << year << endl;
    }
}
```

Python C/C++ Java

Difficulty: **Medium** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Control Structures**
Tags: **Looping Statement** | **Recursion** | **Apply**

Show Testcases Show Solution

Question No. 4

Single File Programming Question

Problem Statement

Program being evaluated will have the responsibility of creating a program that evaluates a truth membership for students. Your task is to develop an interactive program that not only calculates but also showcases the strength of factorial values. Your program should efficiently compute and present the sum of digits for factorial values of any odd numbers within a designated range. This approach will ingeniously keep track of calculations at base, allowing students to derive more intriguing results of mathematics with each answer and details.

Input Format:

The input consists of a single integer, n .

Output Format:

The output displays the factorial and sum of digits of the factorial of odd numbers within the given range.

Refer to the sample output for the formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 ≤ n ≤ 30

Sample Test Cases :

Input 1:

3

Output 1:

1! = 1, sum of digits = 1
3! = 6, sum of digits = 6
5! = 120, sum of digits = 3

Input 2:

11

Output 2:

11! = 39, sum of digits = 3
13! = 6, sum of digits = 6
15! = 132, sum of digits = 3
17! = 5040, sum of digits = 9
19! = 304800, sum of digits = 27
21! = 16931200, sum of digits = 36

Play your code here

```
#include <iostream>
using namespace std;
int fact(int num);
int sum_of_digits(int);
int fact(int digit)
{
    if (digit == 1)
        return 1;
    else
        return digit * fact(digit - 1);
}
int fact(int num)
{
    int sum = 0;
    int digit;
    while (num != 0)
    {
        digit = num % 10;
        sum += sum_of_digits(digit);
        num /= 10;
    }
    return sum;
}
int main()
{
    int start, end;
    cout << "Enter start year : ";
    cin >> start;
    cout << "Enter end year : ";
    cin >> end;
    for (int year = start; year <= end; year++)
    {
        if (year == fact(year))
            cout << year << endl;
    }
}
```

Python C/C++ Java

status: **Solved** **Mark obtained**: **10/10** **Time used**: **0** **Times compiled**: **0** **Times submitted**: **1** **level**: **Medium** **Question type**: **Single File Programming** **Subject**: **Python** **Topic**: **Control Structures**

Sub Topic: **Looping Statement** **Previous Question** **Apply**

Show testcases scores Show solution

Question No. 8

Single File Programming Question

Problem Statement:

Daphne wants to design a program that simulates a real-time scenario based on a mathematical concept known as the Collatz Conjecture. This concept involves the repeated application of rules to a given starting number until the number becomes 1. The rules are as follows:

- If the number is even, divide it by 2.
- If the number is odd, multiply it by 3 and add 1.

Your task is to write a program that takes a positive integer as input, applies the Collatz Conjecture rules to it, counts the number of steps taken to reach 1, and provides an output according to. If the process exceeds 100 steps, the program should print a message indicating an error has occurred.

Input Format:

The input consists of a single integer, n .

Output Format:

The output displays the total number of steps taken to reach 1 from n , under 100. If n is more than 100, it displays "Exceeded 100 steps. Exiting..."

Refer to sample output for the formatting specifications.

Code constraints:

No given test cases fall under the following conditions:
 $1 \leq n \leq 100$

Sample Test Cases:

Input:	Output:
1	Steps taken to reach 1: 1
2	Steps taken to reach 1: 1
3	Exceeded 100 steps. Exiting...

Fill your code here

```
#!/usr/bin/python3
steps=0
while True:
    if steps > 100:
        print("Exceeded 100 steps. Exiting...")
        break
    num = int(input())
    if num % 2 == 0:
        num = num // 2
    else:
        num = num * 3 + 1
    steps += 1
print(f"Steps taken to reach 1: {steps}")

if __name__ == "__main__":
    collatz_steps()
```

status: **Solved** **Mark obtained**: **10/10** **Time used**: **0** **Times compiled**: **0** **Times submitted**: **1** **level**: **Medium** **Question type**: **Single File Programming** **Subject**: **Python** **Topic**: **Control Structures**

Sub Topic: **Looping Statement** **Previous Question** **Apply**

Show testcases scores Show solution

Question No. 8

Single File Programming Question

Problem Statement:

Daphne, a primary school teacher, wants to calculate the sum of numbers within a given range, excluding those that are multiples of 3.

Write a program to help Daphne compute the sum of all numbers between two integer numbers (inclusive) that are not divisible by 3 using the continue statement.

Input Format:

The first line of input consists of an integer, representing the starting number of the range.

The second line of input consists of an integer, representing the ending number of the range.

Output format:

The output prints a single integer, representing the sum of numbers in the range that are not multiples of A .

Refer to the sample output for formating specifications.

Code constraints:

1 ≤ start and end ≤

Sample test cases:**Input 1:**1
10**Output 1:**

33

Input 2:11
20**Output 2:**

77

//Your code here

```
#include<iostream>
int main()
{
    int start, end;
    std::cin >> start >> end;
    int total = 0;
    for (int num = start + 1; num < end; ++num)
        if (num % 3 != 0)
            total += num;
    std::cout << total;
}
```

Status: Solved | My Submissions | 1/100 | 0ms | Unexecuted | 0 | Unsubmitted | 1 | Score: 100 | Question type: Single File Programming | Edit | Programming | Topic: Loops | Sub Topics: Continue |

Block comments: Apply

Show test case scores Show solution

Show testcases scores Show solution

Question No: 3

Single File Programming Question

Problem Statement:

You are practicing programming and it comes about prime number (prime no). We need to write a program that calculates the sum of the non-prime digits in a given integer using loops.

Input file: `sum_of_non_prime_digits.txt`.

Example:

Input:

345

Output:

12

Explanation:

Output: 6 (non-prime), 4 (non-prime), 5 (prime)

The sum of non-prime (say) 3 + 4 = 12

Output: 12

Input format:

The input consists of a single integer.

Output format:

The output prints an integer representing the sum of non-prime digits in it.

Refer to the sample output for formatting specifications.

Code constraints:

0 ≤ n ≤ 10⁹

Sample test cases :

Input 1:

345

Output 1:

12

Input 2:

12345

Output 2:

14

Input 3:

123

Output 3:

1

Fill your code here:

```
#include <stdio.h>
int is_prime(int);
int sum_of_non_prime_digits();
int main()
{
    int sum;
    sum = sum_of_non_prime_digits();
    printf("%d", sum);
}
```

Single File Programming Question

Problem Statement

In 2010, there were 193 countries whose populations had increased over the previous decade. Each country is assigned a unique code based on their population growth rate. Each country is assigned a unique code based on their population growth rate. Each country is assigned a unique code based on their population growth rate. If the code has three specific thresholds, it needs to classify the country's priority level.

Your task is to write a program that takes a country code and its associated data, and then classifies if the priority is "High" or "Low".

Thresholds:

- France: Priority is "High" if the percentage P_1 is > 50, else "Low".
- Japan: Priority is "High" if life expectancy E_1 is > 80, else "Low".
- Brazil: Priority is "High" if the urban population U_1 is > 80, else "Low".

Input Format:

The first line of input consists of an integer, representing the country code (1 for France, 2 for Japan, 3 for Brazil).

If the country code is 1,

the second line contains a floating-point value P_1 , representing the percentage of the total population.

If the country code is 2,

the second line contains a floating-point value E_1 , representing the average life expectancy in years.

If the country code is 3,

the second line contains a floating-point value U_1 , representing the percentage of the urban population.

Output Format:

The first line of output displays "Priority: High" or "Priority: Low" based on the input values.

If the country code is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

Code constraints:

In this segment, the test cases fall under the following constraints:

$1 \leq P_1, E_1, U_1 \leq 100.0$

Sample Test Cases:

Input 1:	Output 1:
1 50.0	Priority: Low
2 85.0	Priority: High
3 75.0	Priority: Low
4 85.0	Priority: Low
5 85.0	Priority: Low
6 85.0	Priority: Low
7 85.0	Priority: Low

My Solution:

```
country = int(input())
if country == 1:
    # France
    print("Priority: High" if P1 >= 50 else "Priority: Low")
elif country == 2:
    # Japan
    print("Priority: High" if E1 >= 80 else "Priority: Low")
elif country == 3:
    # Brazil
    print("Priority: High" if U1 >= 80 else "Priority: Low")
```

Status: Correct | Last submitted: 10 min ago | Unsubmitted | 1 | 1 user | Medium | Guidelines | Single File Programming | Related: Python | Tags: Loops | Due Date: Loops

Show test case scores Show solution

Single File Programming Question

Problem Statement

Students are allowed to work on our computer in our machine room after entering the correct secret code. If the code is correct, the message "Logged In" is displayed. They are not allowed to log in to the machine until they enter the correct secret code.

Write a program to allow the student to work only if he/she enters the correct secret code.

Note: If the secret code starts with the last three digits should be divisible by the first digit of the number.

Input Format:

The input consists of an integer N , which represents the secret code.

Output Format:

The output displays either "Logged In" or "Incorrect code" based on the given condition.

Refer to the sample output for the formatting specifications.

Code Constraints:

No extra spaces; the test cases fall under the following constraints:

$10^4 \leq N \leq 10^5$

Sample Test Cases:

Input 1:	Output 1:
1245	Incorrect code
Input 2:	Output 2:
1245	Logged In

Play your code here

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    if(n % 10 == 0)
        cout << "Logged In";
    else
        cout << "Incorrect code";
}
```

Recommended Learning Content

[#1000 Statement in Python](#)

Status: **Correct** | Mark Obtained: **10/10** | Views: 1 | Times compiled: 1 | Times submitted: 1 | Level: **Medium** | Question type: Single File Programming | Subject: Programming | Tools: Python

Subtopic: **branching and looping** | Last modified: **10/10/2023** | Apply

Show Testcase Codes Show Solution

Student: kamalraj | Email id: 248701225@pjms.edu.in | Test: R00_Python_Week 2_C00_Updated | Course: NeoColabs_R00_CS2321_Python Programming

IP Address: 146.54.114.100 (10.100.1.100) | Last Activity: — | Test Status: Failed (0.00%)

IP Address: 146.54.114.100 (10.100.1.100) | Last Activity: 1 min ago (0.00%)

IP Address: 146.54.114.100 (10.100.1.100) | Last Activity: 1 min ago (0.00%)

IP Address: 146.54.114.100 (10.100.1.100) | Last Activity: 1 min ago (0.00%)

[Summary](#) [Details](#)[Edit](#)[Coding \(0\)](#)

Question No. 1

Single File Programming Question**Problem Statement:**

Given a certain range which is bounded by perfect numbers. A perfect number is a number that equals the sum of its proper divisors (excluding itself). You want to identify all perfect numbers within a given range. Help them write a program to list these numbers.

Input Format:

The first line of input consists of an integer `start`, representing the starting number of the range.
The second line consists of an integer `end`, representing the ending number of the range.

Output Format:

The output prints all perfect numbers in the range, separated by commas.

Refer to the sample output for formatting specifications.

Code constraints :

- 1 ≤ start ≤ 100.
- 1 ≤ end ≤ 100.

Sample test cases :**Input 1:**

```
1
200
```

Output 1:

```
6, 28
```

Input 2:

```
10
20
```

Output 2:

```
10
```

Write your code here:

```
# This is a sample Python script.

def is_perfect(x):
    if x == 1:
        return False
    divisors = [1]
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            divisors.append(i)
            divisors.append(x//i)
    return sum(divisors) == x

start = int(input("Enter start:"))
end = int(input("Enter end:"))

perfect_numbers = []
for num in range(start, end+1):
    if is_perfect(num):
        perfect_numbers.append(str(num))

print(",".join(perfect_numbers))
```

Python 3.8.5

[Status](#) [Cancel](#) [Mark submitted](#) [Run](#) [Reset code](#) [Run as interpreted](#) [Status submitted](#) [Save](#) [Question type](#) Single File Programming [Subject](#) Python [Topic](#) Control Structures

 [Conditional Statement](#) [Infinite Loops](#) [Apply](#)
 [Show testcases](#) [Show solution](#)

Question No. 2

Single File Programming Question**Problem Statement:**

You will be given a simple enrichment program, and you goal is to develop a program that showcases the concept of using control statements to manage loops. Your task is to create a program that takes an integer `N` as input and prints the squares of even numbers from 1 to `N`, while skipping odd numbers.

Input Format:

The input consists of a single integer, which represent the upper limit of the range.

Output Format:

The output displays the square of even numbers from 1 to `N`, separated by commas.

Refer to the sample output for the formatting specifications.

Code constraints:

The given test cases fit under the following constraints:

1 ≤ N ≤ 100

Sample test cases:

Input 1:

10

Output 1:

4
15
30
54
100

Input 2:

21

Output 2:

0
18
36
54
108
134
153

Write your code here

```
#include<iostream.h>
for(1 to range[1] to range[2])
    cout<<range[3]
```

Python... M

Status: **Correct** | Mark obtained: **X/10** | Unmarked: **0** | Time compiled: **1** | Time submitted: **1** | Level: **Easy** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Python Control Statements**

Show testcase scores Show solution

Question No. 3

Single File Programming Question

Problem Statement:

Travis is a mathematics enthusiast. He is picking a range of numbers and wants to count how many of them are not Fibonacci numbers.

Help Travis determine the count of non-Fibonacci numbers within that given range [start, end] using the continue statement.

Input Format:

The first line of input consists of an integer, representing the starting number of the range.

The second line consists of an integer, representing the ending number of the range.

Output Format:

The output prints a single integer, representing the count of numbers in the range that are not Fibonacci numbers.

Refer to the sample output for formatting specifications.

Code constraints:

1 ≤ start < end ≤ 100

Sample test cases:

Input 1:

1
10

Output 1:

0

Input 2:

21
30

Output 2:

0

Write your code here

```
#include<iostream.h>
int generate_Fibonacci(1000);
int main()
{
    int start;
    int end;
    cout<<"Enter start: ";
    cin>>start;
    cout<<"Enter end: ";
    cin>>end;
    generate_Fibonacci(start, end);
}
```

Python... M

```
def main():
    start = int(input())
    end = int(input())
    for num in range(start, end):
        if num % 3 == 0:
            print(num)
```

status: **Completed** | Last submitted: **19:00** | Submissions: **0** | Times compiled: **1** | Errors: **0** | Queue type: Single File Programming | Subject: Programming | Topic: Loops | Sub Topic: Continue

Show test case scores Show solution

Question No. 4

Single File Programming Question

Problem Statement:

As a junior developer working on a text analysis project, your task is to create a program that displays the consonants in a sentence provided by the user, separated by spaces.

You need to implement a program that takes a sentence as input and prints the consonants while skipping vowels and non-alphabetic characters, using only control statements.

Input Format:

The input consists of a string representing the sentence.

Output Format:

One or put multiple space-separated consonants present in the sentence.

Refer to the example output for the formatting specifications.

Code constraints:

In this assignment, you must conform under the following parameters:

The string can include letters [both uppercase and lowercase], punctuation, spaces, and other characters with up to 100 characters.

Sample Test Cases:

Input 1:

Hello world

Output 1:

h l l w r l d

Input 2:

Code

Output 2:

c d

Fill your code here:

Python 3.7

```
# You can write Python
# code here
# Example:
# console.write("HelloWorld")
# console[]

# char_in_sentence:
#     if char_in_sentence >= 'a' and char_in_sentence <= 'z':
#         cons.append(char)
#     print("".join(cons))

# CONSOLE_OUTPUT()
# console.write()
```

status: **Completed** | Last submitted: **19:00** | Submissions: **0** | Times compiled: **1** | Errors: **0** | Queue type: Single File Programming | Subject: Python | Topic: Python Control Statements | Sub Topic: Control Statements | Score: 100 | Create

Show test case scores Show solution

Question No. 5

Single File Programming Question

Problem Statement:

John, a software developer, is analyzing a sequence of numbers within a given range to calculate their digit sum. However, to simplify his task, he includes only numbers that are palindromes. John's task is to find the total (sum) of all digits of all such palindromes.

Note: John finds the total sum of the digits of every palindromic number in the range [start, end] (both inclusive).

Example:

Input

10

20

Output:

10

Explanation:

Range [10, 20]. Only palindromic numbers are 10, 11, 12, 14, 15, 16, 17, 18, 19 and 20.

Digit sum: 1+0 + 1+1 + 1+2 + 1+4 + 1+5 + 1+6 + 1+7 + 1+8 + 1+9 + 2+0 = 35.

Output: 35

Input Format:

The first line of input consists of an integer, representing the starting number of the range.

The second line of input consists of an integer, representing the ending number of the range.

Output Format:

The output prints a single integer, representing the total sum of the digits of all one-palindromic numbers in the range.

Refer to the sample output for formatting specifications.

Code Constraints:

1 ≤ start, end ≤ 100.

Sample Test Cases:

Input 1:

10

20

Output 1:

35

Input 2:

20

21

Output 2:

21

Write your code here

```
#include <iostream>
using namespace std;
int digit_sum(int num);
bool is_palindrome(int num);
int sum_of_all_palindromes_digits(int start, end);
int main()
{
    int sum = sum_of_all_palindromes_digits(start, end);
    cout << sum;
}
```

Source | Contact | Max submitted | **Run** | Refresh | **Times compiled** | **Times submitted** | **Score** | **Body** | Question type: Single File Programming | Subject: Programming | Tags: Loops | Last type: Continue

Show testcases scores Show solution

Student: kamalraj | Email id: 248701225@pjmskhm.edu.in Test: R0C_Python_Week 2_MQ Course: TwoWeeks_R0C_062020_Python Programming

IP Address: 248701225@pjmskhm.edu.in
Last Accessed: 06/06/2020 10:20:50 AMTest Subject: —
Test Start Date: 06/06/2020 10:20:50 AM

Test End Date: 06/06/2020 10:20:50 AM

Recent User: Chrome

Summary Details

More

 0 out of 25

Question No: 11

Multi Choice Type Question

Which keyword is used to immediately terminate a loop?

break ✓

continue

increment

Recommended learning Content: [if else Statement in Python](#)

Status: Correct | Mark obtained: 10 | Weightage: 0 | Level: Easy | SimilarTopics: MCQ Single Correct | Subject: Programming | Topic: Python | Full Topic: Branching and Looping | More Questions | Analysis

 Show Solution

Question No: 12

Multi Choice Type Question

What will be the output of the following python code?

```
1 i = 2
2 while True:
3     if i % 2 == 0:
4         i += 1
5         continue
6     print(i, end = " ")
7     i -= 1
```

155789

155789 ✓

24389

155434378930

Recommended learning Content: [if else Statement in Python](#)

Status: Correct | Mark obtained: 10 | Weightage: 0 | Level: Easy | SimilarTopics: MCQ Single Correct | Subject: Programming | Topic: Python | Full Topic: Branching and Looping | More Questions | Analysis

 Show Solution

Question No: 13

Multi Choice Type Question

How many times will the inner for loop be executed in the below code?

```
1 for i in range(1, 21):
2     for j in range(0, 21):
3         print(i*j)
4         print(" ")
5     if i%2==0:
6         break
```

—

—

—

 0 ✓

1

2

Show solution

Question No:14

Multi Choice Type Question

What will be the output of the following code snippet?

```
1 i = 0
2 while i < 5:
3     if i % 2 == 0:
4         i += 1
5         continue
6     print(i, end=" ")
7     i += 1
```

1 3 4

 1 2 3

1 2 5

1 2 3

 Show solution

Question No:15

Multi Choice Type Question

What will be the output of the following Python code?

```
1 a = 2
2 while True:
3     if a % 3 == 0:
4         a += 2
5         continue
6     if a > 30:
7         break
8     print(a)
9     a += 1
```

 1 2 3 4 5 6 7 8 9 10

1 2 5 7 8 9

1 2 4 8 16

 Show solution

Student: kamalrj | Email id: 248701225@jntuhkcm.edu.in Test: R0C_Python_Week 2_MQ Course: TwoSemester_R0C_062020_Python Programming

 IP Address: 2405-004-000-0000-0000-0000-0000-0000 Test Details TestDuration: 00:00:00 Test Start Date: 08/07/2020 10:22:54 AM TestEnd Test End Date: 08/07/2020 10:22:54 AM Browser Used: Chrome TestEnd TestEnd Date: 08/07/2020 10:22:54 AM TestEnd TestEnd

Question No: 1

Multi Choice Type Question

What is the output of the following?

```
1. x = 2
2. while True:
3.     if 100 == 100:
4.         break
5.     print(1)
6.     x += 2
```

3
4
5
6
72
3

None

 Q 2Recommended Learning Content: [if Statement in Python](#) Status: Done Marked as done Unmarked Level: Medium Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: [Branching and Loops](#) Bloom's Taxonomy: Remember Show solution

Question No: 2

Multi Choice Type Question

What will be the output of the following Python code?

```
1. x = 5
2. while True:
3.     if 100 == 100:
4.         break
5.     print(x, end = " ")
6.     x += 1
```

 Q 3

5 6 7 8 9

 Console / Web browser

3 4

 Status: Wrong Marked as wrong Unmarked Level: Easy Question type: MCQ Single correct Subject: Programming Topic: Python Sub Topic: [Branching and Loops](#) Bloom's Taxonomy: Understood Show solution

Question No: 3

Multi Choice Type Question

What will be the output of the following Python code?

```
1. x = 2
2. while True:
3.     if 100 == 100:
4.         break
5.     print(x)
6.     x += 2
```

None of the above mentioned choice

 Q 4 Q 5

1 2 3 4

Recommended learning content: [If/Elif/Else Statement in Python](#)

Status: Wrong Mark obtained: 0/1 Date created: 0 Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Branching and looping Review history: Understand

Show solution

Question No. 4

Multi Choice Type Question

What will be the output of the following code?

```
for i in range(0):
    if i == 5:
        break
    else:
        print(i)
else:
    print("Hello")
```

0 1 2 3 4 5 Hello

0 1 2 3 4 5 Hello

0 1 2 3 4

1 2 3 4 5

Recommended learning content: [If/Elif/Else Statement in Python](#)

Status: Correct Mark obtained: 1/1 Date created: 0 Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Branching and looping Review history: Understand

Show solution

Question No. 5

Multi Choice Type Question

What will be the output of the following python code?

```
i = 3
while False:
    if i%2 == 0:
        break
    print(i)
    i += 2
```

0 1 2 3 4 5

0 1 2 3 4 5

The code runs successfully but does not print anything

Status: Correct Mark obtained: 1/1 Date created: 0 Question type: MCQ Single Correct Subject: Programming Topic: Python Sub Topic: Strings Review history: Understand

Show solution

Question No. 6

Multi Choice Type Question

What is the purpose of the pass statement in python?

0 To continue a loop without executing the code.

0 To terminate the program.

0 To skip the next line of code.

To do nothing and can be used as a placeholder.

Status: Correct Mark obtained: 1/1 Date created: 0 Question type: MCQ Single Correct Review history: Understand

Show solution

Question No. 7

Multi Choice Type Question

What will be the output of the following python code?

```
1 i = 3
2 while True:
3     if i == 0:
4         break
5     print(i)
6     i += 1
```

3 6 9 12

3 6

A. None

B. 3 6 9

Status: Wrong | MarkedIncorrect: 0 | Attempted: 0 | Level: Medium | QuestionType: MCQ Single Correct | Subject: Python | Topic: Loops | SubTopic: Loops | PreviousQuestion: NextQuestion: Understood

Show solution

Question No. 8

Multi Choice Type Question

What will be the output for the following code snippet?

```
1 i = 0
2 for i in range(10):
3     break
4     print(i)
```

None of the mentioned options

1

0

D. 0

Status: Correct | MarkedIncorrect: 0 | Attempted: 0 | Level: Easy | QuestionType: MCQ Single Correct | Subject: Python | Topic: Control Structures | SubTopic: Loops Statement | PreviousQuestion: NextQuestion: Understood

Show solution

Question No. 9

Multi Choice Type Question

What will be the output of the following code?

```
1 i = 1
2 while True:
3     if i == 1:
4         i -= 1
5     print(i)
6     i += 1
```

None of the mentioned options

1 2 3 4 5 6 7

None of the mentioned options

C. 1 2 3 4 5 6 7

Recommended learning Content: [if else Statement in Python](#)

Status: Correct | MarkedIncorrect: 0 | Attempted: 0 | Level: Medium | QuestionType: MCQ Single Correct | Subject: Programming | Topic: Python | SubTopic: Branching and Loops | PreviousQuestion: NextQuestion: Understood

Show solution

Question No. 10

Multi Choice Type Question

What does the else statement written other than the loop execute?

A. While loop condition becomes false

None of the mentioned options

The statement is always executed

When break statement is executed in the loop

Recommended learning Content: [if else Statement in Python](#)

Status: Correct | MarkedIncorrect: 0 | Attempted: 0 | Level: Easy | QuestionType: MCQ Single Correct | Subject: Programming | Topic: Python | SubTopic: Branching and Loops | PreviousQuestion: NextQuestion: Understood

Show solution

1 2 3 4

Student: kamalraj | Email id: 248701225@psuokhml.edu.in Test: R00_Python_Week1_PA1 Course: RealData_R00_C03021_Python Programming

IP Address: 248701225@PSUOKHML4666:16550:207.46.173.165
Last Submited: 08-03-2020 10:22:59 AM

Test Status: Failed | Run: 0.0000 | Result: 0.0000

Dif Used: 0.0000
Last Submited: 08-03-2020 10:22:59 AMDif Used: 0.0000
Last Submited: 08-03-2020 10:22:59 AM

Summary Details

Coding (W)

More

Question No: 1

Single File Programming Question**Problem Statement:**

A smart home system tracks the temperature and humidity of each room. Create a program that takes the room name (string), temperature (float), and humidity (float) and displays the room's climate details.

Input Format:

The first line of input contains a string representing the room name.
 The second line contains a float value representing the temperature.
 The third line contains a float value representing the humidity.

Output Format:

The first line of output prints "Room: " followed by the room name (string).
 The second line prints "Temperature: " followed by the temperature (float) formatted to two decimal places.
 The third line prints "Humidity: " followed by the humidity (float) formatted to two decimal places and a percentage sign (%).

Refer to the sample output for formatting specifications.

Code constraints :0.0 < Temperature < 100.0
0.0 < Humidity < 100.0**Sample Test Cases :****Input 1:**Living Room
21.45
45.78**Output 1:**Room: Living Room
Temperature: 21.45
Humidity: 45.78%**Input 2:**Kitchen
30.12
38.29**Output 2:**Room: Kitchen
Temperature: 30.12
Humidity: 38.29%

Fill your code here

```
a=input()
b=input()
c=input()
print("Room: ",a)
print("Temperature: ",b)
print("Humidity: ",c,"%")
```

Status: **Current** Block submitted: **0** Review queued: **0** Review submitted: **0** Last: Medium Question type: Single File Programming Author: Python Topic: Datatypes Sub Topic: Datatypes

Show previous scores Show summary

Question No: 2

Single File Programming Question**Problem Statement:**

Mandy is debating with her friend Rachel about an interesting mathematical claim. Rachel asserts that for any positive integer n , the ratio of the sum of n odd integers to the integer $f(n)$ is always 4. Introducing this by the statement, decide to validate it using logical operators and basic arithmetic.

This question is available if you purchased both the previous questions.

Input format:

The input consists of a positive integer N , representing the integer to be tested.

Output format:

The first line of output displays "Sum" followed by an integer representing the calculated sum.
The second line displays "Rachit's statement is" followed by a Boolean value indicating whether Rachit's statement is correct.

Refer to the sample output for the formatting specifications.

Code constraints:

In this section, the test cases fall under the following constraints:

$1 \leq N \leq 1000$

Sample Test Cases:

Input 1:	Output 1:
1	Sum: 40 Rachit's statement is: True
Input 2:	Output 2:
70	Sum: 390 Rachit's statement is: True

```
print("Sum: 40,Rachit's statement is: True")
```

Status: Correct Mark obtained: 10 Attempt count: 0 Status submitted: 3 Level: Hard Question type: Single File Programming Subject: Python Topic: Operators SubTopic: Operators.

Show testcases scores Show solutions

Question No. 3**Single File Programming Question****Problem Statement**

Itman, a passionate baker, is planning to bake cookies for a large party via original recipe created by his mother, with the following ingredient quantities: 0.5 cup of flour, 1 cup of sugar, and 0.6 cup of butter.

Write a program to calculate the amounts of flour, sugar, and butter needed for a certain number of cookies. Provide the required quantities for a specific number of cookies, maintaining the original proportion of the recipe.

Input format:

The input consists of an integer N , representing the number of cookies.

Output format:

The first line prints "Flour: X cups" where X represents the amount of flour required to maintain the original value rounded to two decimal places.

The second line prints "Sugar: Y cups" where Y represents the amount of sugar required to maintain the original value rounded to two decimal places.

The third line prints "Butter: Z cups" where Z represents the amount of butter required to maintain the original value rounded to two decimal places.

Refer to the sample output for formatting specifications.

Code constraints:

In this section, the test cases fall under the following constraints:

$1 \leq N \leq 100$

Sample Test Cases:

Input 1:	Output 1:
5	Flour: 2.50 cups Sugar: 2.00 cups Butter: 3.00 cups
Input 2:	Output 2:
10	Flour: 5.00 cups Sugar: 4.00 cups Butter: 6.00 cups
Input 3:	Output 3:

Score: 10.0% (max)
Solved: 1,871 / 1991
Difficulty: 3.25 - easy

Input 4:

10

Output 4:

Flour: 2.25 cups
Sugar: 1.50 cups
Butter: 0.50 cups

#Your code here

```
# This one uses functions
def calculate():
    flr=2.25
    sgr=1.50
    btr=0.50
    flr+flr
    sgr+sgr
    btr+btr
    print("Flour: ",flr,"cups", "Sugar: ",sgr)
    print("Butter: ",btr,"cups", "Total: ",flr+sgr+btr)
```

System.out.println()

Recommended Learning Content: Python Data Operators Selected in Python

Status: Solved Mark Milestone: Refreshed: 4 Times compiled: 3 Times submitted: 3 Level: Medium Question Type: Single File Programming Subject: Programming Tools: Python Full Topic Operators

 Show test case scores Show solution**Question No. 4****Single File Programming Question****Problem Statement**

John works at a car dealership and is responsible for recording the details of cars that come in the showroom. To make his job easier, he wants a program that can take the car's make, model, and price, and display the information in a formatted manner.

Instructions to the program:

Input Format:

The first line of input contains a string representing the car make.
The second line contains a string, representing the car model.

The third line contains a float value, representing the car price.

Output Format:

The first line of output prints "Car Model", followed by the car make.
The second line prints "Car Model", followed by the car model.
The third line prints "Price", followed by the car price, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Code constraints :

0.0 ≤ price ≤ 100000.0

Sample Test Cases :**Input 1:**

```
Toyota
Corolla
2000.75
```

Output 1:

```
Car Model: Toyota
Car Model: Corolla
Price: 20,000.75
```

Input 2:

```
BMW
Model 3
2000.00
```

Output 2:

```
Car Model: BMW
Car Model: Model 3
Price: 20,000.00
```

#Your code here

```
# This one uses functions
def input():
    flr=2.25
    sgr=1.50
    btr=0.50
    print("Car Model: ",flr,"Car Model: ",sgr,"Car Model: ",btr)
```

System.out.println()

Status: **Correct** - Last submitted: 1h ago - Run count: 6 - Times compiled: 3 - Times submitted: 4 - Level: Medium - Question type: Single File Programming - Subject: Python - Tags: Datatypes, List Topic: Operators

Show testcases scores Show solution

Question ID: 5

Single File Programming Question

Problem Statement:

You can only fit one episode in a single viewing session for now. Once there is a specific number of times after watching a set number of episodes, the time is short break.

She is provided with the following information:

- 1. Each episode of the series has a fixed duration of 45 minutes.
- 2. After a certain number of episodes, there is a break of 10 minutes.

She wants to know the total time she will need to watch the entire series, including the breaks. Your task is to help her by calculating the total viewing time.

Input Format:

The first line of input consists of an integer n , representing the total number of episodes in the series.

The second line consists of an integer k , representing the number of episodes watched before taking a break.

Output Format:

The output prints an integer representing the total viewing time required to watch the entire series, including the breaks.

Refer to the sample output for formatting specifications.

Code constraints:

In this question, the test cases fall under the following constraints:

1 ≤ n ≤ 20

1 ≤ k ≤ 10

Sample Test Cases:

Input 1: 4 2	Output 1: 195 minutes
Input 2: 5 3	Output 2: 215 minutes
Input 3: 6 4	Output 3: 235 minutes

If you need help:

```
def main():
    n = int(input())
    k = int(input())
    total_watching_time = 0
    total_breaks = 0
    for i in range(1, n+1):
        if i > k:
            total_watching_time += 45
            total_breaks += 10
        else:
            total_watching_time += 45
    print(total_watching_time + total_breaks)
```

Recommended Learning Content: [Python Data Types](#) [Operators in Python](#)

Status: **Correct** - Last submitted: 1h ago - Run count: 6 - Times compiled: 3 - Times submitted: 4 - Level: Medium - Question type: Single File Programming - Subject: Programming - Tags: Python, List Topic: Operators

Show testcases scores Show solution

Question No. 6

Single File Programming Question

Problem Statement:

Cesar is planning to movie night with his friends and wants to download a high-definition movie. He knows the file size of the movie in megabytes (MB) and his internet speed in megabits per second (Mbps). To ensure the movie is ready by 9PM, Cesar needs to calculate the download time.

The task is to write a program that calculates the download time and displays it in hours, minutes, and seconds.

Example:

Input:
MB = 800
Mbps = 40

Output:

Download Time: 0 hours, 2 minutes, and 40 seconds.

Explanation:

- Convert the file size to bits ($800 \text{ MB} * 8 \text{ megabit} = 6400 \text{ megabit}$) and divide it by the download speed ($6400 \text{ megabit} / 40 \text{ Mbps} = 160 \text{ seconds}$).
- Now, convert the download time in seconds to hours, minutes, and seconds. 160 seconds is equal to 2 minutes and 40 seconds.

So, the download time is 0 hours, 2 minutes and 40 seconds.

Input Format:

The first line of input consists of an integer N , representing the file size in megabytes (MB).
The second line consists of an integer S , representing the network speed in megabits per second (Mbps).

Output Format:

You output prints "Download Time: H Hours, M Minutes, and S Seconds", where H, M, and S are integers representing the hours, minutes, and seconds respectively.

Refer to the sample output for formatting specifications.

Code Constraints:

In this scenario, the test cases fall under the following constraints:

1 <= N <= 10⁴

1 <= S <= 40

Sample Test Cases:

Input 1:

```
800
40
```

Output 1:

Download Time: 0 hours, 2 minutes, and 40 seconds.

Input 2:

```
600
30
```

Output 2:

Download Time: 0 hours, 8 minutes, and 32 seconds.

Input 3:

```
1000
50
```

Output 3:

Download Time: 0 hours, 2 minutes, and 40 seconds.

Write your code here:

```
#include <iostream>
using namespace std;
int main()
{
    int N, S;
    cin >> N >> S;
    cout << "Download Time: " << N << " MB, " << S << " Mbps" << endl;
    cout << "Download Time: " << N * 8 << " bits, " << S << " Mbps" << endl;
    cout << "Download Time: " << N * 8 / S << " seconds" << endl;
    cout << "Download Time: " << N * 8 / S / 60 << " minutes, " << N * 8 / S % 60 / 60 << " hours, " << N * 8 / S % 60 % 60 / 60 << " seconds" << endl;
    cout << "Download Time: " << N * 8 / S % 60 % 60 << " seconds" << endl;
}
```

Recommended Learning Content:

Python Data Structures

Operated in Python

Status: **Success** | Mark obtained: **10** | Attempted: **1** | Times compiled: **1** | Times submitted: **2** | Level: **Hard** | Category: **Single File Programming** | Subject: **Programming** | Tools: **Python** | Sub Topic: **Operations**

Review accuracy: **Apply**

Show testcases codes Show solution

Student: kamalrj | Email: 248701225@projdatashm.edu.in Test: R00_Python_Week1_CV Course: NeoSrib_R00_OG3321_Python Programming

IP Address: 248701225@97346060344001248701225.vt... | Test Status: —

IP Address: 94.237.104.166:8000/10223.PDF

IP Address: 94.237.104.166:8000/10223.PDF

IP Address: 94.237.104.166:8000/10223.PDF

[Summary](#) [Testcases](#)[More](#)[Coding \(4\)](#)

Question No: 1

Single File Programming Question**Problem Statement:**

You are an air traffic controller who needs to record and manage flight delays efficiently. Given a flight number, the time in minutes (0 to nudge), and the coordinates of the flight's current position (in a complex number), help Air control record this information in a structured format.

Input Format:

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

The second line contains a string representing the delay in minutes.

The third line consists of two floats separated by a space, representing the real and imaginary parts of the complex number for the flight's position.

Output Format:

The first line of output displays the complex number.

The second line displays a string with the flight number, delay, and the real and imaginary parts of the complex number, separated by commas.

Refer to the sample output for formatting specifications.

Code constraints:

$1 \leq n \leq 100000$

$0 \leq \text{delay} \leq 1000$

$-100000 \leq \text{real}, \text{im} \leq 100000$

Sample Test Cases:**Input 1:**

```
12345
30.5
32.3 45.8
```

Output 1:

```
(32.3+45.8j)
12345, 30.5, 32.3, 45.8
```

Input 2:

```
12345
30.5
32.3 45.8
```

Output 2:

```
(32.3+45.8j)
12345, 30.5, 32.3, 45.8
```

Fill your code here:

```
#!/usr/bin/python3
n = int(input())
delay = float(input())
real, img = map(float, input().split())
print(complex(real, img))
print(delay)
print(f"({n},{delay},{real},{img})")
```

Status: **Correct** | Mark obtained: **10/10** | Time used: **0** | Test cases passed: **0** | Test cases failed: **0** | Level: **Medium** | Question type: **Single File Programming** | Subject: **Python** | Topic: **DataTypes** | Full Topic: **DataTypes**

Show testcases source | Show testcases

Question No: 2

Single File Programming Question**Problem Statement:**

Erlly is organizing a house-party and needs to determine the total number of tacos required and the total cost. Each attendee at the party will consume 2 tacos. To ensure there are enough tacos:

If there are 10 or more attendees, Erlly will need to purchase an additional 5 tacos.

If there are fewer than 10 attendees, Erlly must purchase an additional 2 tacos and consider

The cost of each taco is \$10. Write a program that calculates both the total number of tacos required and the total cost based on the number of attendees.

Input Format:

The input consists of one integer representing the number of attendees.

Output Format:

The first line prints "Number of tacos needed:" followed by an integer representing the number of tacos needed in attendees.

The second line prints "Total cost:" followed by an integer representing the total cost.

Refer to the sample output for the formatting specifications.

Code constraints:

In this exercise, you must consider the following constraints:

1 ≤ n ≤ 100

The cost per taco is fixed at \$10.

Each attendee consumes exactly 2 tacos.

Sample Test Cases:

Input 1:

10

Output 1:

Number of tacos needed: 20

Total cost: 200

Input 2:

100

Output 2:

Number of tacos needed: 200

Total cost: 2000

Input 3:

1

Output 3:

Number of tacos needed: 2

Total cost: 20

My code here:

```
#!/usr/bin/python3
n=10
print("Number of tacos needed: ",n)
print("Total cost: ",n*10)
```

Python... ▾

Status: **correct** | My solution: **8/10** | Attempted: **0** | Times compiled: **4** | Times submitted: **1** | Level: **Medium** | Essential tools: **Single File Programming** | Subject: **Python** | Topic: **operators** | Author: **operators**

Results history: **Copy**

Show testcases scores Show solution

Overview Note 3

Single File Programming Question

Problem Statement

John is working on a project involving multiple servers. Each server provides a very small value that needs to be processed to compute an aggregated value.

Given two integers n_1 and n_2 , write a program to calculate the aggregated value using specific bitwise operations and arithmetic computations. The final result should be the aggregated value modulo 1000.

Example:

Input:
1 1
1 2
2 1
2 2

Output:

1

Explanation:

Compute the bitwise AND of servers 1 and servers 2 (mod 1000).
Calculate the XOR of this result from step 1 and servers 3 (mod 1000).
Multiply the result from step 2 by 3 (mod 1000).
Compute the final aggregated value by taking the result from step 3 modulo 1000. So, the aggregated value is 1.

Input Format:

The first line of input consists of an integer n_1 representing server 1 data.

The second line of input consists of an integer n_2 representing server 2 data.

The third line of input consists of an integer n_3 representing server 3 data.

Output Format:

The output displays an integer representing the aggregated result.

Refer to the sample output for the formatting specifications.

Code constraints:

In this scenario, the test cases fall under the following constraints:
1 ≤ n ≤ 100, 0 ≤ k ≤ 40000

Sample test cases:

Input 1:

```
1  
2  
3
```

Output 1:

```
1  
2  
3
```

Input 2:

```
1  
2  
3
```

Output 2:

```
1  
2  
3
```

Write your code here

```
a = int(input())
b = int(input())
c = int(input())
d = ((a+b)*c)/10000
print(d)
```

Recommended Learning Classes: Python Interview Questions

Important in Python

Status: **Unsolved** Mark Submitted: 0/00 Time used: 0 Status categorized: 0 Status submitted: 0 Level: Hard Category: Single File Programming Subject: Programming Topic: Python Sub Topic: Operators

Show Testcase scores Show solution

Question No: 4

Single File Programming Question

Problem Statement:

Given a user's dashboard for his new fitness app interface, she needs a program that can capture and display key details about a user's workout. The program should read the user's information, the total steps they can take during their workout in kilometers, and the duration of their workout in hours. After collecting this information, the program will generate a summary of the user's fitness activity.

Your task is to guide them through the program.

Input Format:

The first line of input consists of a string, representing the user's name.
The second line consists of an integer, representing the total steps taken.
The third line consists of a float value, representing the calories burned.
The fourth line consists of a float value, representing the workout duration in hours.

Output Format:

The first line of output prints "User Name:" followed by the user's name.
The second line prints "Total Steps:" followed by the total steps.
The third line prints "Calories Burned:" followed by the calories burned, rounded off to one decimal place.
The fourth line prints "Workout duration: <hours>" where <hours> is the workout duration, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

Code constraints:

The name is a non-empty string.
1 ≤ total steps ≤ 10000
0.0 ≤ calories burned ≤ 10000.0
0.0 ≤ workout duration ≤ 100.0

Sample test cases:

Input 1:

```
John  
30000  
350.5  
1.1
```

Output 1:

```
User Name: John  
Total Steps: 30000  
Calories Burned: 350.5  
Workout Duration: 1.1 hours
```

Input 2:

```
Jenny  
25000
```

Output 2:

```
User Name: Jenny  
Total Steps: 25000
```

88.1
1.1Calculator Score: 88.1
Heroku Duration: 1.1 hours

Edit your code here

```
# Your code using Python
name = input()
print(len(name))
calculator = float(input())
duration = float(input())
info = input().split(" ")
print("Name: " + info[0])
print("Phone: " + info[1])
print("Calculator: " + str(calculator))
print("Duration: " + str(duration))
```

Python... ▾

Status: **Success** | Last submitted: **10/10** | Times used: **0** | Times compiled: **00** | Times submitted: **2** | Level: **Hard** | Question type: **Single File Programming** | Subject: **Python** | Topic: **Decomposition** | Full page | Edit types

Success summary: **Apply**

Show test case scores Show solution

Student: kamalraj | Email id: 248701225@pjstech.edu.in Test: R00_Python_Week1_C00 Course: NeoColab_R00_C033021_Python Programming

IP Address: 2405:684:208:1E54:600:914:4:200

Tim Srinivas

Dhaval Patel

Bhavin Patel, Chrome

Summary Details

More

Coding (0)

Question No: 1

Single File Programming Question**Problem Statement:**

John, the owner of a popular bakery, wishes to create a special offer code for his customers. To generate this code, he plans to combine the day of the month with the number of items left in stock.

Help him to encode these two values into a unique offer code.

Note: Use the bitwise operator to calculate the offer code.

Example**Input:**

22

Output:

Offer code: 15

Explanation:

Given the day of the month 22 & day inventory 15 and there are 0 items left (Inventory 100), the offer code is returned as 15, which is 11.

Input Format:The first line of input consists of an integer D representing the Day Of The Month.The second line consists of an integer I representing the number of items left in stock.**Output Format:**

You must display "Offer code" followed by an integer representing the encoded offer code.

Refer to the sample output for formatting specifications.

Code constraints:

In this session, the test cases fall under the following constraints:

1 ≤ D ≤ 301 ≤ I ≤ 500**Sample Test Cases:****Input 1:**

22

15

Output 1:

Offer code: 15

Input 2:

22

10

Output 2:

Offer code: 10

(No code here)

```
# You will write Python
def OfferCode(D,I):
    return D^I
```

Python 3.7.4

Status: Correct | Last updated: 10 minutes ago | Status computed: 1 minute ago | Module: NeoColab | Question type: Single File Programming | Subject: Python | Topic: Operators | Sub Topic: Operators

Score summary: **Analysis** Show testcasescores Show solution

Question No: 2

Single File Programming Question

Problem Statement:

A science experiment produces a decimal value as the result. However, the scientist needs to convert this value into an integer so that it can be used in further calculations.

Write a Python program that takes a floating-point number as input and converts it into an integer.

Input Format:

The input consists of a floating-point number t .

Output Format:

The output prints "The integer value of t is: [result]", followed by the integer variable equivalent to the floating-point number.

Refer to the sample output for the formatting specifications.

Code constraints:

$1.0 \leq t \leq 10^2$

Sample test cases:

Input:	Output 1:	Output 2:
10.35	The integer value of 10.35 is: 10	The integer value of 1.351 is: 1

Write your code here

```
# This code is being run in
# an online compiler
# Don't edit
print("The integer value of 10.35 is: 10")
```

Status: Solved | Mark obtained: 10 | Time used: 4 | Times compiled: 1 | Times submitted: 1 | View | Save | Question type: Single File Programming | Subject: Python | Topic: Data Types | Sub-Topic: Data Types

Show testcases Show solution

Question No. 3

Single File Programming Question

Problem Statement:

A company has hired two employees, Alice and Bob. The company wants to swap the salaries of both employees. Alice's salary is an integer value and Bob's salary is a floating-point value.

Write a program to swap their salaries and print the new salary of each employee.

Input Format:

The first line of input consists of an integer A , representing Alice's salary.

The second line consists of a float B , representing Bob's salary.

Output Format:

The first line of output displays "Alice's salary".

The second line displays "Alice's salary = A ", where A is Alice's salary.

The third line of output displays "Bob's salary = B ", where B is Bob's salary.

After a new line space, it is followed by the message "New salaries after swapping".

The next line displays "Alice's salary = V ", where V is the swapped salary.

The last line displays "Bob's salary = V ", where V is the swapped salary.

Refer to the sample output for formatting specifications.

Code constraints:

$1.0 \leq A \leq 10^2$

$1.0 \leq B \leq 100000.0$

Sample test cases:

Input:	Output 1:	Output 2:
10000 15488.0	Initial salaries: Alice's salary = 10000 Bob's salary = 15488.0	New salaries after swapping: Alice's salary = 15488.0 Bob's salary = 10000.0

Input 2:

```
20000  
20000.0
```

Output 2:

```
Initial salaries:  
Alice's salary = 20000  
Bob's salary = 20000.0  
  
New salaries after mapping:  
Alice's salary = 20000.0  
Bob's salary = 20000.0
```

(View code here)

```
# You are writing Python  
# A) Input C)  
# B) Input C)  
# C) Python  
# D) Python  
# E) Java  
# F) C++  
# G) C  
# H) C#  
# I) Python  
# J) Java  
# K) C#  
# L) Python  
# M) Java  
# N) C#  
# O) Python  
# P) Java  
# Q) C#  
# R) Python  
# S) Java  
# T) C#  
# U) Python  
# V) Java  
# W) C#  
# X) Python  
# Y) Java  
# Z) C#
```

Submit Cancel Check plagiarism View suggested Save Question type: Single File Programming Subject: Python Topic: Datatypes Last Update: Datatypes

Show testable codes Show solution

Question No. 4

Single File Programming Question

Problem Statement

In a family, two children receive allowances based on the gardening tasks they complete. The older child receives an allowance rate of Rs.5 per task, with a bonus allowance of Rs.10. The younger child receives an allowance rate of Rs.3 for each task, with a bonus allowance of Rs.5.

Your task is to calculate and display the allowances for the older and younger children based on the number of gardening tasks they complete, along with the total allowances for both children combined.

Input format:

The first line of input displays "Older child allowances: ", followed by an integer representing the number of tasks completed by the older child.
The second line displays "Younger child allowances: ", followed by an integer representing the number of tasks completed by the younger child.

Output format:

The first line of output displays "Older child allowances: ", followed by an integer representing the allowances calculated for the older child.
The second line displays "Younger child allowances: ", followed by an integer representing the allowances calculated for the younger child.
The third line displays "Total allowances: ", followed by an integer representing the sum of both child's allowances.

Refer to the sample output for formatting specifications.

Code constraints :

The given test cases fall under the following constraints:

1 <= n <= 30

Sample test cases :

Input 1:

```
20  
10
```

Output 1:

```
Older child allowances: Rs.100  
Younger child allowances: Rs.30  
Total allowances: Rs.130
```

Input 2:

```
20  
25
```

Output 2:

```
Older child allowances: Rs.100  
Younger child allowances: Rs.75  
Total allowances: Rs.175
```

Input 3:

```
20  
30
```

Output 3:

```
Older child allowances: Rs.100  
Younger child allowances: Rs.90  
Total allowances: Rs.190
```

(View code here)

```
# You are writing Python  
# A) Python  
# B) Python  
# C) Python  
# D) Python  
# E) Python  
# F) Python  
# G) Python  
# H) Python  
# I) Python  
# J) Python  
# K) Python  
# L) Python  
# M) Python  
# N) Python  
# O) Python  
# P) Python  
# Q) Python  
# R) Python  
# S) Python  
# T) Python  
# U) Python  
# V) Python  
# W) Python  
# X) Python  
# Y) Python  
# Z) Python
```

[Create](#) [Import](#) [Delete](#) [Edit](#) [Print](#) [CSV](#) [XLS](#) [PDF](#) [Email](#) [Report](#) [Module](#) [Question type](#) [Single-line Programming](#) [Subject](#) [Python](#) [Topic](#) [Operations](#) [Sub-Topic](#) [Operations](#)

Disney Store® cookie jar. ©Disney. All rights reserved.

Quarantine Area 1

Single File Programming Question

Problem Statement

Given a natural number n , by exploring the properties of numbers, find the sum of all consecutive integers, considering the averages of their fourth powers and their subtracting the product of the first and last consecutive integers, constant value.

To estimate my hypothesis, what if the result is mixed consistent and theory

Querido
input:

Output:

61 of 64

1. Calculate $a^4 - b^4 = 625$
 2. Calculate $(a+1)^4 - b^2 = 1296$
 3. Calculate $(a+2)^4 - b^4 = 2401$
 4. Calculate $(a+3)^4 - b^2 = 4096$

Method of analysis

Brownman [520] + 5029 + 2401 + 403000/4 = 238457

Received, from LUDWIGSBURG THE PRELIMINARY PRODUCT OF A COMPOUND + 20

Wavelength = $\lambda = (\lambda_0 + \Delta\lambda) = 500 + 50 = 550$

Fitted results: 330435 - 40 = 326435

Input format

The legal aspects of decisions of representing the law of his country also include

Survey format

This modified oligoacrylate ("Crosslinked polyacrylate") softens and then crosslinks acrylic hydrogels in Quinton's formula.

Refer to the [current output for formatting specifications](#).

Codel constraints:

In this section, the first comes full under the following constraints:

1443

Somnile test cross:

Input 1

Input 3:-

```
10) your code here
```

Status: **Correct** | Last submitted: **10 minutes ago** | Status updated: **10 minutes ago** | **Run** | **Save** | Classifier type: **Single file Programming** | **Submit** | **Python** | Topic: **Operations** | **All Topics** | **Questions**

Recent history: **analyze**

Show test case scores Show solution

Student: kamalrj | Email: 248701225@pjms.edu.in Test: POC_Python_Week1_MQ Course: NeoData_POC_C63023_Python Programming

 IP Address: 140.50.149.193 (Cloudflare) Test Details: [View Test Details](#) Test Date: 2023-08-01 10:00:00 UTC+05:30 Test Status: [View Test Status](#) Device Used: Windows Test Duration: 00:00:00.000000000 Device Used: Chrome Test Duration: 00:00:00.000000000Summary Details

Edit

 100%

Question No: 11

Multi Choice Type Question

Which of the following can convert the string to a float number?

 float()

float(123)

int(float(12))

int(123)

Recommended learning Content: [Data Types and Variables](#)

Status: Correct | Mark obtained: 10 | Time used: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub Topic: Operators | Round Robbin: Random | Underload:

 Show Solution

Question No: 12

Multi Choice Type Question

What will be the output of the following code?

```
i=3 * 24**12/132-45/36  
print(i)
```

 -72

-64

-60

-600

Status: Correct | Mark obtained: 10 | Time used: 0 | Level: Medium | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub Topic: Operators | Round Robbin: Random | Underload:

Question No: 13

Multi Choice Type Question

What will be the value of the following Python expression?

$$4 \times 23/5$$
 9 7 5 4Recommended learning Content: [Python Bitwise Operators](#) [Operators in Python](#)

Status: Correct | Mark obtained: 10 | Time used: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub Topic: Operators | Round Robbin: Random | Underload:

Question No: 14

Multi Choice Type QuestionThe value of the expression $4(3^4(3-1))$ and $4(3^4(3-1))$ is the same. True or False? False

Q True

status: correct | Mark obtained: 10 | Attempted: 6 | Level: Easy | Question type: MCQ Single correct | Subject: Python | Topic: Operators | Sub-Topic: Operators | Exam category: Analysis

Question No.15

Multi Choice Type Question

What of the following represents the bitwise NOT operator?

A.

B.

C.

D.



ANSWER

status: Correct | Mark obtained: 10 | Attempted: 6 | Level: Easy | Question type: MCQ Single correct | Subject: Python | Topic: Operators | Sub-Topic: Operators | Exam category: Analysis

Show solution

1 2 3

Student: kamalrj | Email: 248701225@projektauhm.edu.in Test: POC_Python_Week1_MCQ Course: NeoData_POC_C63023_Python Programming

 IP Address: 240.104.191.125 (Cloudflare) Test Device: — Device Used: Windows Test Duration: 00:00:00.000 (00:00:00) Test Duration Since Last Test: 00:00:00.000 (00:00:00) Device Used: Chrome Test Duration Since Last Test: 00:00:00.000 (00:00:00)Summary

More

 0/4 (0%)

Question No. 1

Multi Choice Type Question

Indicate the exponent operation if no answer is 16.
 A. 8
 B. 16
 C. 32
 D. 0 ✓

Recommended Learning Content: Python Bitwise Operators | Operators In Python

Status: Correct | Mark Obtained: 10 | Time used: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub Topic: Operators | Scored Recovery: Understood

Question No. 2

Multi Choice Type Question

What is used to concatenate two strings in Python?

A. operator
 B. separator
 C. separator ✓

None of the mentioned options

Status: Correct | Mark Obtained: 10 | Time used: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub Topic: String | Scored Recovery: Understood

Show solution

Question No. 3

Multi Choice Type Question

What is the value of the following expression?
 A. 12.0
 B. 12.00
 C. 12.000
 D. 12.0000 ✓

Recommended learning content: Python Bitwise Operators | Operators In Python

Status: Correct | Mark Obtained: 10 | Time used: 0 | Level: Easy | Question type: MCQ Single Correct | Subject: Programming | Topic: Python | Sub Topic: Operators | Scored Recovery: Understood

Question No. 4

Multi Choice Type Question

What is the value of the following expression?
 A. 20
 B. 200
 C. 2000
 D. 20000 ✓

Q 4

Question No. 5

Multi Choice Type Question

What of the following functions converts a string to a float in Python?

O `int()`

`int([base])`

`long([base])`

Show solution

Question No. 6

Multi Choice Type Question

Which of these is not a core data type?

Dictionary

Tuple

O Class

List

Show solution

Question No. 7

Multi Choice Type Question

What is the output of the following number conversion?

`z = complex(1,2)`

`print(z)`

O 1+2j

1-2j

ValueError: Missing an imaginary part of a complex number

1j

Show solution

Question No. 8

Multi Choice Type Question

What must 3 * 4 evaluate to?

12

78

-3.75

Q. 7

Status: **Correct** Mark obtained: **10** Revision count: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **Data types** Sub-topic: **Operators** Previous question Next question Analysis

Show solution

Question No: 8

Multi Choice Type Question

What will be the output of the following code?

1. `x = [3, 5, 1, 2 * 3]; print(x)`

33

35

43

D. 33



Status: **Correct** Mark obtained: **10** Revision count: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **Data types** Sub-topic: **Operators** Previous question Next question Analysis

Question No: 10

Multi Choice Type Question

What will be the following code output?

1. `a = 3 + 4; print(a[1])`

D. 44



33

35

40

Status: **Correct** Mark obtained: **10** Revision count: **0** Level: **Easy** Question type: **MCQ Single Correct** Subject: **Python** Topic: **Data types** Sub-topic: **Operators** Previous question Next question Analysis

Show solution