

# Rajalakshmi Engineering College

Name: Sushanth Reddy P V  
Email: 241901115@rajalakshmi.edu.in  
Roll no: 241901115  
Phone: 9840125574  
Branch: REC  
Department: I CSE (CS) FB  
Batch: 2028  
Degree: B.E - CSE (CS)

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_MCQ

Attempt : 1  
Total Mark : 20  
Marks Obtained : 17

#### Section 1 : MCQ

1. What will be the output?

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

**Answer**

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

**Status : Correct**

**Marks : 1/1**

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

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

```
print(c)
print(tuple(c))
```

**Answer**

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

**Status : Correct**

**Marks : 1/1**

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

**Answer**

```
()
```

**Status : Correct**

**Marks : 1/1**

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

**Answer**

Removes the key-value pair for the key given as an argument

**Status : Wrong**

**Marks : 0/1**

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

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

**Answer**

False

**Status : Correct**

**Marks : 1/1**

6. What is the output of the following code?

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

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

**Answer**

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

**Status :** Correct

**Marks :** 1/1

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

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

**Answer**

13

**Status :** Correct

**Marks :** 1/1

8. What is the output of the following code?

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

**Answer**

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

**Status :** Correct

**Marks :** 1/1

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

**Answer**

False

**Status :** Correct

**Marks :** 1/1

10. Set  $s_1 = \{1, 2, 4, 3\}$  and  $s_2 = \{1, 5, 4, 6\}$ , find  $s_1 \cap s_2$ ,  $s_1 - s_2$ ,  $s_1 \cup s_2$

s2 and s1 ^ s2.

**Answer**

s1&s2 = {1, 4}s1-s2 = {2, 3}s1^s2 = {2, 3, 5, 6}s1|s2 = {1, 2, 3, 4, 5, 6}

**Status : Correct**

**Marks : 1/1**

11. What is the output of the following code?

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

**Answer**

False

**Status : Correct**

**Marks : 1/1**

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

**Answer**

t[3] = 45

**Status : Correct**

**Marks : 1/1**

13. Which of the following is a Python tuple?

**Answer**

(1, 2, 3)

**Status : Correct**

**Marks : 1/1**

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

```
list1= [1, 2, 3]  
list2 = [5, 6, 7]  
list3 = [10, 11, 12]
```

```
set1 = set(list2)
set2 = set(list1)
set1.update(set2)
set1.update(list3)
print(set1)
```

**Answer**

{1, 2, 3, 5, 6, 7, 10, 11, 12}

**Status :** Correct

**Marks :** 1/1

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

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

**Answer**

{1, 2, 3}

**Status :** Correct

**Marks :** 1/1

16. What is the output of the following?

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

**Answer**

TrueFalse

**Status :** Wrong

**Marks :** 0/1

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

**Answer**

When duplicate keys encountered, the last assignment wins

**Status :** Wrong

**Marks :** 0/1

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

**Answer**

Values of a dictionary must be unique

**Status :** Correct

**Marks :** 1/1

19. Predict the output of the following Python program

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

**Answer**

{1, 2, 7, 8}TypeError: unsupported operand type

**Status :** Correct

**Marks :** 1/1

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

Output:

Tuple: (1, 3, 4)

Max value: 4

t=(1,)

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

**Answer**

1)  $t = t + (3, 4) \cdot 2 \cdot \max(t)$

Status : Correct

Marks : 1/1

# Rajalakshmi Engineering College

Name: Sushanth Reddy P V  
Email: 241901115@rajalakshmi.edu.in  
Roll no: 241901115  
Phone: 9840125574  
Branch: REC  
Department: I CSE (CS) FB  
Batch: 2028  
Degree: B.E - CSE (CS)

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_COD

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

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

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

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

Example



Input:

6 //number of product ID

101

102

101

103

101

102 //product IDs

Output:

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

Total Unique IDs: 3

Average Frequency: 2.00

Explanation:

Input 6 indicates that you will enter 6 product IDs.

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

Input 101: Added with a frequency of 1.

Input 102: Added with a frequency of 1.

Input 101: Frequency of 101 increased to 2.

Input 103: Added with a frequency of 1.

Input 101: Frequency of 101 increased to 3.

Input 102: Frequency of 102 increased to 2.

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

Total Unique is 3.

The average frequency is 2.00.

**Input Format**

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

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

### ***Output Format***

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

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

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

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 6

101

102

101

103

101

102

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

Total Unique IDs: 3

Average Frequency: 2.00

### ***Answer***

# You are using Python

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

```
frequency = {}
```

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

```
    product_id = int(input())
```

```
    if product_id in frequency:
```

```
        frequency[product_id] += 1
```

```
else:
    frequency[product_id] = 1
total_unique = len(frequency)
average_frequency = sum(frequency.values()) / total_unique
print(frequency)
print(f"Total Unique IDs: {total_unique}")
print(f"Average Frequency: {average_frequency:.2f}")
```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

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

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

### ***Input Format***

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

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

### ***Output Format***

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

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 2

101 100 150 200

102 50 75 100

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

**Answer**

# You are using Python

```
def analyze_sales_data():
```

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

```
    sales_data={}
```

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

```
        data=list(map(int, input().split()))
```

```
        customer_id=data[0]
```

```
        amounts=data[1:]
```

```
        total_spent=sum(amounts)
```

```
        max_expenditure=max(amounts)
```

```
        sales_data[customer_id]=[total_spent,max_expenditure]
```

```
    print(sales_data)
```

```
analyze_sales_data()
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

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

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

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

### ***Input Format***

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

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

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

### ***Output Format***

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

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

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 1 2 3

2 3 4

3 4 5

Output: {2, 3}

{2}

### ***Answer***

# You are using Python

```
def analyze_par():  
    register=set(map(int, input().split()))  
    attend=set(map(int, input().split()))  
    drop=set(map(int ,input().split()))  
    attend_re=register&attend  
    print(attend_re)  
    stay=attend_re - drop  
    print(stay)
```

analyze\_par()

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

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

Write a program to help Gowshik with this task.

Example:

Given list:

[1, 2, 3, 4]

[3, 5, 2, 1]

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

#### ***Input Format***

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

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

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

#### ***Output Format***

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

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 4

1, 2, 3, 4

3, 5, 2, 1

Output: (4, 7, 5, 5)

### **Answer**

# You are using Python

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

```
l1=list(map(int,input().split(',')))
```

```
l2=list(map(int,input().split(',')))
```

```
sum_tuple=tuple(a+b for a,b in zip(l1,l2))
```

```
print(sum_tuple)
```

**Status :** Correct

**Marks :** 10/10

## **5. Problem Statement**

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

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

**Note:**

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

### **Input Format**

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

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

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

### **Output Format**

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

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 2

(1, [1, 2])

(2, [3, 4])

2

Output: 3 4

### **Answer**

# You are using Python

```
def filter_q(items,threshold):
```

```
    result=list(filter(lambda x:x >threshold,[quantity for item in items for quantity in item[1]]))
```

```
    return result
```

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

```
items=[]
```

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

```
    item=eval(input())
```

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

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

```
filtered_q=filter_q(items,threshold)
```

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

**Status : Correct**

**Marks : 10/10**



# Rajalakshmi Engineering College

Name: Sushanth Reddy P V  
Email: 241901115@rajalakshmi.edu.in  
Roll no: 241901115  
Phone: 9840125574  
Branch: REC  
Department: I CSE (CS) FB  
Batch: 2028  
Degree: B.E - CSE (CS)

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_PAH

Attempt : 1  
Total Mark : 60  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

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

Help Jordan in implementing the same.

#### ***Input Format***

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

#### ***Output Format***

The output prints a single integer formed by concatenating the unique integers

from the input in the order they appeared.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 11 11 33 50

Output: 113350

### **Answer**

```
# You are using Python
# Read space-separated integers from input
numbers = input().split()

# Use an ordered set approach to remove duplicates while preserving order
unique_numbers = []
seen = set()

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

# Concatenate the unique integers into a single string and print the result
print("".join(unique_numbers))
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

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

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

### **Input Format**

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

### **Output Format**

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

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

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

### **Answer**

# You are using Python

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

```
def generate_prime_dict(n):  
    prime_dict = {}  
    num = 2  
    count = 1
```

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

```
    return prime_dict
```

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

```
print(generate_prime_dict(n))
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

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

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

#### **Input Format**

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

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

#### **Output Format**

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

Refer to the sample output for formatting specifications.

#### **Sample Test Case**

Input: 3

5 10 15

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

#### **Answer**

```
# You are using Python
n=int(input())
```

```
l=list(map(int,input().split()))
t=[]
for i in range(n):
    if i<n-1:
        t.append(tuple((l[i],l[i+1])))
    else:
        t.append(tuple((l[i],None)))
print(tuple(t))
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

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

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

##### ***Input Format***

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

##### ***Output Format***

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

Refer to the sample output for formatting specifications.

##### ***Sample Test Case***

Input: 5

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

##### ***Answer***

```
# You are using Python
def generate_square_dict(n):
    return {i: i**2 for i in range(1, n+1)}
```

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

```
print(generate_square_dict(n))
```

**Status :** Correct

**Marks :** 10/10

## 5. Problem Statement

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

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

### ***Input Format***

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

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

### ***Output Format***

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

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 3

1  
2  
3

Output: (1, 2, 3)

**Answer**

# You are using Python

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

```
l=[]
```

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

```
    s=int(input())
```

```
    l.append(s)
```

```
l.sort()
```

```
groups=[]
```

```
cur=[l[0]]
```

```
for i in range(1,n):
```

```
    if l[i]==l[i-1]+1:
```

```
        cur.append(l[i])
```

```
    else:
```

```
        groups.append(tuple(cur))
```

```
        cur=[l[i]]
```

```
groups.append(tuple(cur))
```

```
for i in groups:
```

```
    if(len(i)==1):
```

```
        print('(' +str(i[0])+')')
```

```
    else:
```

```
        print(i)
```

**Status : Correct**

**Marks : 10/10**

## 6. Problem Statement

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

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

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

the choice is invalid.

### ***Input Format***

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

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

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

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

### ***Output Format***

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

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

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

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

For invalid choices: Print "Invalid choice".

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 1 2 3 4 5

1

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

5

### ***Answer***

# You are using Python



```
# Read initial set
initial_numbers = list(map(int, input().split()))
number_set = set(initial_numbers)

# Read user choice
ch = int(input())

# Print the original set in descending order as per sample output
sorted_desc = sorted(number_set, reverse=True)
print("{ " + ", ".join(str(x) for x in sorted_desc) + "}")

# Perform operations based on choice
if ch == 1:
    print(max(number_set))
elif ch == 2:
    print(min(number_set))
elif ch == 3:
    n1 = int(input())
    number_set.discard(n1) # removes if present, no error if not
    sorted_desc = sorted(number_set, reverse=True)
    print("{ " + ", ".join(str(x) for x in sorted_desc) + "}")
else:
```

**Status : Wrong**

**Marks : 0/10**