Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 36.5

Section 1 : Coding

1. Problem Statement

Alex is creating an account and needs to set up a password. The program prompts Alex to enter their name, mobile number, chosen username, and desired password. Password validation criteria include:

Length between 10 and 20 characters.At least one digit.At least one special character from !@#$%^&amp;* set. Display "Valid Password" if criteria are met; otherwise, raise an exception with an appropriate error message.

Input Format

The first line of the input consists of the name as a string.

The second line of the input consists of the mobile number as a string.

The third line of the input consists of the username as a string.

The fourth line of the input consists of the password as a string.

Output Format

If the password is valid (meets all the criteria), it will print "Valid Password"

If the password is weak (fails any one or more criteria), it will print an error message accordingly.

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: John
9874563210
john
john1#nhoj

Output: Valid Password

Answer

```
import re

def validate_password(password):
    if not any(char.isdigit() for char in password):
        return "Should contain at least one digit"

    if not any(char in "!@#$%^&*" for char in password):
        return "It should contain at least one special character"

    if not (10 <= len(password) <= 20):
        return "Should be a minimum of 10 characters and a maximum of 20
characters"

    return "Valid Password"

name = input().strip()
mobile_number = input().strip()
username = input().strip()
password = input().strip()
```

```
print(validate_password(password))
```

Status : Partially correct ***Marks : 6.5/10***

2. Problem Statement

Bob, a data analyst, requires a program to automate the process of analyzing character frequency in a given text. This program should allow the user to input a string, calculate the frequency of each character within the text, save these character frequencies to a file named "char_frequency.txt," and display the results.

Input Format

The input consists of the string.

Output Format

The first line prints "Character Frequencies:".

The following lines print the character frequency in the format: "X: Y" where X is the character and Y is the count.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: aaabbbccc

Output: Character Frequencies:
a: 3
b: 3
c: 3

Answer

```
from collections import OrderedDict

def analyze_character_frequency(text):
    frequencies = OrderedDict()
    for char in text:
        frequencies[char] = frequencies.get(char, 0) + 1

    with open("char_frequency.txt", "w") as file:
        file.write("Character Frequencies:\n")
        for char, count in frequencies.items():
            file.write(f"{char}: {count}\n")

    print("Character Frequencies:")
    for char, count in frequencies.items():
        print(f"{char}: {count}")

text = input().strip()
analyze_character_frequency(text)
```

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

3. Problem Statement

In the enchanted realm of Academia, you, the Academic Alchemist, are bestowed with a magical quill and a parchment to weave the grades of aspiring students into a tapestry of academic brilliance.

The mission is to craft a Python program that empowers faculty members to enter student grades for any two subjects, stores these magical grades in a mystical file, and then, with a wave of your virtual wand, 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 saved in a mystical file named "magical_grades.txt".

Refer to the sample output for format specifications.

Sample Test Case

Input: Alice
Math
95
English
88
done

Output: 91.50

Answer

```
def record_grades():
    with open("magical_grades.txt", "w") as file:
        while True:
            student_name = input().strip()
            if student_name.lower() == "done":
                break

            subject1 = input().strip()
            grade1 = input().strip()
            subject2 = input().strip()
            grade2 = input().strip()

            try:
                grade1 = float(grade1)
                grade2 = float(grade2)

                if not (0 <= grade1 <= 100 and 0 <= grade2 <= 100):
                    print("Grades should be between 0 and 100")
                    continue

                gpa = (grade1 + grade2) / 2
                file.write(f"{student_name}, {subject1}: {grade1}, {subject2}: {grade2}\n")
                print(f"{gpa:.2f}")

            except ValueError:
```

```
            print("Invalid grade input. Please enter numerical values.")
record_grades()
```

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

4. 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 that, 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 HH:MM:SS'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.

Sample Test Case

Input: 2022-01-12 06:10:00
2022-02-12 10:10:12

Output: 2022-01-12 06:10:00
2022-02-12 10:10:12

Answer

```
from datetime import datetime

def validate_datetime(input_str):
    try:
        dt = datetime.strptime(input_str, "%Y-%m-%d %H:%M:%S")
        return dt.strftime("%Y-%m-%d %H:%M:%S")
    except ValueError:
        return "Event time is not in the format"

start_time = input().strip()
end_time = input().strip()

valid_start = validate_datetime(start_time)
valid_end = validate_datetime(end_time)

if valid_start == "Event time is not in the format" or valid_end == "Event time is
not in the format":
    print("Event time is not in the format")
else:
    print(valid_start)
    print(valid_end)
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_PAH

Attempt : 1
Total Mark : 30
Marks Obtained : 26

Section 1 : Coding

1. Problem Statement

John is a data analyst who often works with text files. He needs a program that can analyze 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 'X' appears {Y} times in the file." where X is the character and Y i 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.

Sample Test Case

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

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

Answer

```
filename = input()
content = input()
char_to_count = input()


with open(filename, "w") as file:
    file.write(content)


with open(filename, "r") as file:
    file_content = file.read()


count = file_content.lower().count(char_to_count.lower())


if count > 0:
    print(f"The character '{char_to_count}' appears {count} times in the file.")
```

```
else:
   print("Character not found in the file.")
```

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

2. Problem Statement

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

If a duplicate or a 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 30 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 each contain two space-separated values, representing 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. Error: Student ID already exists."

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

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 3
16 Sam
87 Sabari
43 Dani

Output: Student with ID 16 added to the database.
Student with ID 87 added to the database.
Student with ID 43 added to the database.

Answer

```
class StudentNotFoundException(Exception):
    """Exception raised when a student is not found in the database."""
    pass

class StudentDuplicateIdException(Exception):
    """Exception raised when trying to add a student with an ID that already
exists."""
    def __init__(self):
        self.message = "Student ID already exists."
        super().__init__(self.message)

class StudentDatabaseFullException(Exception):
    """Exception raised when trying to add a student to a full database."""
    def __init__(self):
        self.message = "Student database is full."
        super().__init__(self.message)

class Student:
    """Class representing a student with an ID and name."""
    def __init__(self, student_id, name):
        self.student_id = student_id
        self.name = name

class StudentDatabase:
    """Class for managing a database of students."""
    MAX_CAPACITY = 30

    def __init__(self):
        self.students = []
    def add_student(self, student):
        """
        Add a student to the database.

        Args:
            student: The Student object to be added.

        Raises:
            StudentDuplicateIdException: If a student with the same ID already exists.
            StudentDatabaseFullException: If the database has reached its maximum
capacity.
        """
        # Check if database is full
        if len(self.students) >= self.MAX_CAPACITY:
            raise StudentDatabaseFullException()

        # Check for duplicate ID
        for existing_student in self.students:
            if existing_student.student_id == student.student_id:
                raise StudentDuplicateIdException()

        # Add student to database
        self.students.append(student)
        print(f"Student with ID {student.student_id} added to the database.")

    def get_student(self, student_id):
        """
        Get a student from the database by ID.

        Args:
            student_id: The ID of the student to retrieve.

        Returns:
            The Student object with the given ID.

        Raises:
            StudentNotFoundException: If no student with the given ID exists.
        """
        for student in self.students:
            if student.student_id == student_id:
                return student
        raise StudentNotFoundException(f"No student found with ID {student_id}")

    def get_all_students(self):
        """
        Get all students in the database.

        Returns:
            A list of all Student objects in the database.
        """
        return self.students

# Main function to handle input and process students
def main():
    database = StudentDatabase()

    # Read the number of students to add
    n = int(input().strip())

    # Process each student
    for _ in range(n):
        # Read student ID and name
        student_input = input().strip().split(" ", 1)
        student_id = int(student_input[0])
        student_name = student_input[1] if len(student_input) > 1 else ""

        # Create student object
        student = Student(student_id, student_name)

        try:
            # Try to add the student to the database
            database.add_student(student)
        except (StudentDuplicateIdException, StudentDatabaseFullException) as e:
            print(f"Exception caught. Error: {e.message}")

if __name__ == "__main__":
    main()
```

Status : *Partially correct* ***Marks : 6/10***

3. Problem Statement

Reeta is playing with numbers. Reeta wants to have a file containing a list of numbers, and 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 (as 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: X.XX" (where X.XX 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.

Sample Test Case

Input: 1 2 3 4 5

Output: Average of the numbers is: 3.00

Answer

```
input_line = input()
values = input_line.split()
numbers = []

try:
   for value in values:
      numbers.append(float(value))
   average = sum(numbers) / len(numbers)
   print(f"Average of the numbers is: {average:.2f}")
except ValueError:
   print("Invalid data in the input.")
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_COD

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

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

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

File Name: sentence_file.txt

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.

Sample Test Case

Input: Four Words In This Sentence
Output: 5

Answer

```
sentence = input()


with open("sentence_file.txt", "w") as file:
    file.write(sentence)


with open("sentence_file.txt", "r") as file:
    content = file.read()


words = content.strip().split()
word_count = len(words)

print(word_count)
```

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

2. 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 RuntimeError with the message: "Excessive Cost".

Input Format

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

The second line consists of 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.

Sample Test Case

Input: 20.0
5

Output: 100.0

Answer

```
try:

    price = float(input())
    quantity = int(input())


    if price <= 0:
        raise ValueError("Invalid Price")


    if quantity <= 0:
        raise ValueError("Invalid Quantity")
    total_cost = price * quantity


    if total_cost > 1000:
        raise RuntimeError("Excessive Cost")


    print(f"{total_cost:.1f}")

except ValueError as ve:
    print(ve)
except RuntimeError as re:
    print(re)
```

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

3. 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.

Sample Test Case

Input: 18
Output: Eligible to vote

Answer

```
class NotEligibleToVote(Exception):
    pass


age = int(input())

try:
    if age < 18:
        raise NotEligibleToVote
    else:
        print("Eligible to vote")
except NotEligibleToVote:
    print("Not eligible to vote")
```

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

4. 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: text_file.txt

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.

Sample Test Case

Input: #SpecialSymBoLs1234

Output: Original String: #SpecialSymBoLs1234
Upper-Case String: #SPECIALSYMBOLS1234
Lower-Case String: #specialsymbols1234

Answer

```
user_input = input()


with open("text_file.txt", "w") as file:
   file.write(user_input)

with open("text_file.txt", "r") as file:
   original_string = file.read()


upper_case_string = original_string.upper()
lower_case_string = original_string.lower()


print(f"Original String: {original_string}")
print(f"Upper-Case String: {upper_case_string}")
print(f"Lower-Case String: {lower_case_string}")
```

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

5. 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 a 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 an element 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.

Sample Test Case

Input: -2
1
2
Output: Error: The length of the list must be a non-negative integer.

Answer

```
try:
    n_input = input()
    n = int(n_input)

    if n <= 0:
        print("Error: The length of the list must be a non-negative integer.")
    elif n > 20:
        print("Error: The length of the list must not exceed 20.")
    else:
        total = 0
        count = 0
        for _ in range(n):
            try:
                num_input = input()
                num = int(num_input)
                total += num
                count += 1
            except ValueError:
                print("Error: You must enter a numeric value.")
                break
        else:
            average = total / count
            print(f"The average is: {average:.2f}")
except ValueError:
    print("Error: You must enter a numeric value.")
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_MCQ

Attempt : 1
Total Mark : 20
Marks Obtained : 16

Section 1 : MCQ

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

Answer

raise Exception()

Status : Correct *Marks : 1/1*

2. Fill in the blanks in the following code of writing data in binary files.

```
import _____________ (1)
rec=[]
while True:
  rn=int(input("Enter"))
  nm=input("Enter")
  temp=[rn, nm]
```

```
    rec.append(temp)
    ch=input("Enter choice (y/N)")
    if ch.upper=="N":
        break
f.open("stud.dat","____________")(2)
__________.dump(rec,f)(3)
_________.close()(4)
```

Answer

(pickle,wb,pickle,f)

Status : Correct ***Marks : 1/1***

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

Answer

The finally block is always executed, regardless of whether an exception occurs or not

Status : Correct ***Marks : 1/1***

4. What is the output of the following code?

```
try:
    x = "hello" + 5
except TypeError:
    print("Type Error occurred")
finally:
    print("This will always execute")
```

Answer

Type Error occurredThis will always execute

Status : Correct ***Marks : 1/1***

5. Which of the following is true about

fp.seek(10,1)

Answer

Move file pointer ten characters ahead from the current position

Status : Correct ***Marks : 1/1***

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

Answer

error

Status : Wrong ***Marks : 0/1***

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

file_object.seek(offset [,reference_point])

Answer

0

Status : Correct ***Marks : 1/1***

8. How do you rename a file?

Answer

os.set_name(existing_name, new_name)

Status : Wrong ***Marks : 0/1***

9. What is the output of the following code?

```
try:
   x = 1 / 0
except ZeroDivisionError:
   print("Caught division by zero error")
finally:
   print("Executed")
```

Answer

Caught division by zero errorExecuted

Status : Correct *Marks : 1/1*

10. What will be the output of the following Python code?

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

Answer

True

Status : Correct *Marks : 1/1*

11. What will be the output of the following Python code?

```
# Predefined lines to simulate the file content
lines = [
    "This is 1st line",
    "This is 2nd line",
    "This is 3rd line",
    "This is 4th line",
    "This is 5th line"
]

print("Name of the file: foo.txt")

# Print the first 5 lines from the predefined list
for index in range(5):
    line = lines[index]
    print("Line No %d - %s" % (index + 1, line.strip()))
```

Answer

Displays Output

Status : Correct ***Marks : 1/1***

12. What is the purpose of the except clause in Python?

Answer

To handle exceptions during code execution

Status : Correct ***Marks : 1/1***

13. Match the following:

a) f.seek(5,1) i) Move file pointer five characters behind from the current position

b) f.seek(-5,1) ii) Move file pointer to the end of a file

c) f.seek(0,2) iii) Move file pointer five characters ahead from the current position

d) f.seek(0) iv) Move file pointer to the beginning of a file

Answer

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

Status : Wrong ***Marks : 0/1***

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

Answer

The program will exit automatically

Status : Wrong ***Marks : 0/1***

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

Answer

By creating a new class that inherits from the Exception class

Status : Correct ***Marks : 1/1***

16. Fill the code to in order to read file from the current position.

Assuming exp.txt file has following 3 lines, consider current file position is beginning of 2nd line

Meri,25

John,21

Raj,20

Ouptput:

['John,21\n','Raj,20\n']

```
f = open("exp.txt", "w+")
__________(1)
print _________(2)
```

Answer

1) f.seek(0, 1)2) f.readlines()

Status : Correct ***Marks : 1/1***

17. What is the difference between r+ and w+ modes?

Answer

in r+ the pointer is initially placed at the beginning of the file and the pointer is at the end for w+

Status : Correct ***Marks : 1/1***

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

Output:

Name of the file: ex.txt

```
fo = open(______(1), "wb")
print("Name of the file: ",______)(2)
```

Answer

1) "ex.txt"2) fo.name

Status : Correct ***Marks : 1/1***

19. What is the output of the following code?

```
class MyError(Exception):
   pass

try:
   raise MyError("Something went wrong")
except MyError as e:
   print(e)
```

Answer

Something went wrong

Status : Correct ***Marks : 1/1***

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

Answer

finally

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 37.5

Section 1 : Coding

1. Problem Statement

James is an engineer working on designing a new rocket propulsion system. He needs to solve a quadratic equation to determine the optimal launch trajectory. The equation is of the form ax2 +bx+c=0.

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

Input Format

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

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

Output Format

The output displays:

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
1 5 6

Output: (-2.0, -3.0)

Answer

```
import cmath

def find_roots():
    # Read input
    N = int(input())  # Number of coefficients (should be 3)
    if N != 3:
        print("Invalid input")
        return

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

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

    if discriminant > 0:
        # Two distinct real roots
        root1 = (-b + discriminant**0.5) / (2 * a)
        root2 = (-b - discriminant**0.5) / (2 * a)
        print((root1, root2))
    elif discriminant == 0:
        # One repeated real root
        root = -b / (2 * a)
        print((root,))
    else:
        # Complex roots
        real_part = -b / (2 * a)
        imaginary_part = cmath.sqrt(discriminant) / (2 * a)
        print(((real_part, imaginary_part.imag), (real_part, -imaginary_part.imag)))

# Run the program
find_roots()
```

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

2. Problem Statement

Samantha is working on a text analysis tool that compares two words to find common and unique letters. She wants a program that reads two words, w1, and w2, and performs the following operations:

Print the letters common to both words, in alphabetical order.Print the letters that are unique to each word, in alphabetical order.Determine if the set of letters in the first word is a superset of the letters in the second word.Check if there are no common letters between the two words and print the result as a Boolean value.

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

Your task is to help Samantha in implementing the same.

Input Format

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

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

Output Format

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

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

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: program
Peace
Output: ['a', 'p']
['c', 'e', 'g', 'm', 'o', 'r']
False
False
```

Answer

```
def analyze_words():
    # Read input and clean up spaces and case differences
    w1 = input().strip().lower()
    w2 = input().strip().lower()

    # Convert words to sets of letters
    set_w1 = set(w1)
    set_w2 = set(w2)

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

    # Find unique letters (sorted)
    unique_letters = sorted((set_w1 | set_w2) - (set_w1 & set_w2))
    print(unique_letters)

    # Check if w1 is a superset of w2
    print(set_w1.issuperset(set_w2))
    # Check if there are no common letters
    print(set_w1.isdisjoint(set_w2))

# Run the program
analyze_words()
```

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

3. Problem Statement

Riya owns a store and keeps track of item prices from two different suppliers using two separate dictionaries. He wants to compare these prices to identify any differences. Your task is to write a program that calculates the absolute difference in prices 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 original prices.

Help Riya to implement the above task using a dictionary.

Input Format

The first line of input consists of an integer n1, representing the number of items in the first dictionary.

The next n1 lines contain two integers

1. The first line contains the item (key), and
2. The second line contains the price (value).

The following line consists of an integer n2, representing the number of items in the second dictionary

The next n2 lines contain two integers

1. The first line contains the item (key), and
2. The second line contains the price (value).

Output Format

The output should display a dictionary that includes:

1. For items common to both dictionaries, the absolute difference between their prices.
2. For items that are unique to one dictionary, the original price from that dictionary.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
4
4
1
8
7

Output: {4: 4, 8: 7}

Answer

```
def compare_prices():
    # Read the number of items in first dictionary
    n1 = int(input().strip())

    # Initialize first dictionary
    dict1 = {}
    for _ in range(n1):
        key = int(input().strip())
        value = int(input().strip())
        dict1[key] = value

    # Read the number of items in second dictionary
    n2 = int(input().strip())

    # Initialize second dictionary
    dict2 = {}
    for _ in range(n2):
        key = int(input().strip())
        value = int(input().strip())
        dict2[key] = value
    # Initialize result dictionary
    result = {}

    # Process items in first dictionary
    for key, value in dict1.items():
        if key in dict2:
            # If key exists in both dictionaries, calculate absolute difference
            result[key] = abs(value - dict2[key])
        else:
            # If key is unique to dict1, include with original price
            result[key] = value

    # Process items that are unique to second dictionary
    for key, value in dict2.items():
        if key not in dict1:
            # If key is unique to dict2, include with original price
            result[key] = value

    # Display the result
    print(result)

# Call the function to execute the program
compare_prices()
```

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

4. Problem Statement

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

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

Compute the difference between the borrowed books (first set) and the returned books (second set).Compute the difference between the added books (third set) and the missing books (fourth set).Find the union of the results from the previous two steps, and sort the final result in descending order.

Input Format

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

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

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

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

Output Format

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

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3
2 3 4
5 6 7
6 7 8

Output: [1]
[5]
[5, 1]

Answer

```
def main():
    # Reading input sets
    borrowed_books = set(map(int, input().split()))  # Set P
    returned_books = set(map(int, input().split()))  # Set Q
    added_books = set(map(int, input().split()))     # Set R
    missing_books = set(map(int, input().split()))   # Set S

    # Step 1: Difference between borrowed books and returned books
    borrowed_not_returned = borrowed_books - returned_books
    # Step 2: Difference between added books and missing books
    added_not_missing = added_books - missing_books
    # Step 3: Union of the two results
    final_result = borrowed_not_returned | added_not_missing

    # Printing the results
    print(sorted(borrowed_not_returned, reverse=True))
    print(sorted(added_not_missing, reverse=True))
    print(sorted(final_result, reverse=True))

# Run the program
main()
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_PAH

Attempt : 1
Total Mark : 60
Marks Obtained : 60

Section 1 : Coding

1. Problem Statement

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

```
def group_consecutive_event_ids(n, event_ids):
    # Sort the event IDs to ensure they are in ascending order
    event_ids.sort()

    # Initialize an empty list to store the groups
    groups = []

    # Temporary list to hold a group of consecutive IDs
    current_group = [event_ids[0]]

    # Iterate through the event IDs to form consecutive groups
    for i in range(1, n):
        if event_ids[i] == event_ids[i - 1] + 1:
            # If the current ID is consecutive, add it to the group
            current_group.append(event_ids[i])
        else:
            # If the current ID is not consecutive, finalize the previous group
            groups.append(tuple(current_group))
            current_group = [event_ids[i]]  # Start a new group

    # Append the last group
    groups.append(tuple(current_group))

    # Print the groups as tuples, adjusting for single-element tuples
    for group in groups:
        if len(group) == 1:
            print(f"({group[0]})", end=" ")
        else:
            print(group, end=" ")
    print()  # To ensure the output ends with a newline

def main():
    # Read the number of event IDs
    n = int(input())

    # Read the event IDs into a list
    event_ids = [int(input()) for _ in range(n)]

    # Call the function to group and print the consecutive event IDs
    group_consecutive_event_ids(n, event_ids)

# Run the program
if __name__ == "__main__":
    main()
```

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

2. 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

```
def main():
    # Step 1: Read the input and split the integers
    input_numbers = input().split()

    # Step 2: Initialize a set to keep track of seen numbers
    seen = set()
    unique_numbers = []

    # Step 3: Loop through the input numbers
    for num in input_numbers:
        if num not in seen:
            unique_numbers.append(num)  # Add to result if not seen
            seen.add(num)  # Mark this number as seen

    # Step 4: Concatenate the unique numbers into a single string
    result = ''.join(unique_numbers)

    # Step 5: Print the result
    print(result)

# Run the program
if __name__ == "__main__":
    main()
```

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

3. 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

```
def generate_square_dict(n):
    # Generate dictionary using dictionary comprehension
    square_dict = {i: i**2 for i in range(1, n + 1)}

    # Print the dictionary
    print(square_dict)

def main():
    # Read the input number n
    n = int(input())

    # Call the function to generate and print the square dictionary
    generate_square_dict(n)

# Run the program
if __name__ == "__main__":
    main()
```

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

4. 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

```
def main():
    # Step 1: Read the input
    n = int(input())  # Read the number of elements in the list
    elements = list(map(int, input().split()))  # Read the space-separated integers
as a list

    # Step 2: Generate the pairs
    result = []
    for i in range(n - 1):
        result.append((elements[i], elements[i+1]))  # Pair consecutive elements

    # Add the last pair with None
    result.append((elements[-1], None))

    # Step 3: Print the result
    print(tuple(result))  # Print the result as a tuple of pairs

# Run the program
if __name__ == "__main__":
    main()
```

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

5. 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

```
def main():
    # Read input
    numbers = set(map(int, input().split()))
    choice = int(input())

    # Print the original set in descending order
    sorted_numbers = sorted(numbers, reverse=True)
    print("{" + ", ".join(map(str, sorted_numbers)) + "}")

    # Perform operations based on user choice
    if choice == 1:
        print(max(numbers))
    elif choice == 2:
        print(min(numbers))
    elif choice == 3:
        n1 = int(input())
        if n1 in numbers:
            numbers.remove(n1)
        sorted_numbers = sorted(numbers, reverse=True)
        print("{" + ", ".join(map(str, sorted_numbers)) + "}")
    else:
        print("Invalid choice")

# Run the program
main()
```

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

6. 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

```
import math

# Function to check if a number is prime
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(math.sqrt(num)) + 1):
        if num % i == 0:
            return False
    return True

# Function to generate the first n primes and store them in a dictionary
def generate_primes(n):
    primes = {}
    num = 2
    count = 1

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

    return primes

# Main function to read input and display the output
def main():
    # Step 1: Read the input value
    n = int(input())  # the number of primes

    # Step 2: Generate the dictionary of prime numbers
    result = generate_primes(n)

    # Step 3: Print the resulting dictionary
    print(result)

# Run the program
if __name__ == "__main__":
    main()
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_COD

Attempt : 1
Total Mark : 50
Marks Obtained : 46

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

```
from collections import defaultdict

def main():
    # Read number of product IDs
    n = int(input())

    # Initialize a dictionary to track the frequency of each product ID
    freq = defaultdict(int)

    # Read the product IDs and update their frequencies
    for _ in range(n):
        product_id = int(input())
        freq[product_id] += 1

    # Calculate the total number of unique product IDs
    unique_ids = len(freq)

    # Calculate the average frequency of product IDs
    total_frequency = sum(freq.values())
    avg_frequency = total_frequency / unique_ids if unique_ids != 0 else 0

    # Output the frequency dictionary
    print(f"{dict(freq)}")

    # Output the total unique product IDs
    print(f"Total Unique IDs: {unique_ids}")

    # Output the average frequency rounded to two decimal places
    print(f"Average Frequency: {avg_frequency:.2f}")

# Run the main function
if __name__ == "__main__":
    main()
```

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

```
def main():
    # Read the number of customers
    n = int(input())

    # Initialize an empty dictionary to store results
    customer_data = {}

    # Read data for each customer
    for _ in range(n):
        # Read the input for a customer
        data = list(map(int, input().split()))

        # Extract customer ID and the amounts they spent
        customer_id = data[0]
        amounts = data[1:]

        # Calculate the total expenditure and the highest single expenditure
        total_expenditure = sum(amounts)
        max_expenditure = max(amounts)

        # Store the results in the dictionary
        customer_data[customer_id] = [total_expenditure, max_expenditure]

    # Print the dictionary as required
    print(customer_data)

# Run the main function
if __name__ == "__main__":
    main()
```

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

```
def main():
   # Read the three sets of student IDs
   registered = set(map(int, input().split()))
   attended = set(map(int, input().split()))
   dropped_out = set(map(int, input().split()))

   # Find the students who registered and attended (intersection of registered
and attended)
   registered_and_attended = registered.intersection(attended)

   # Find the students who registered, attended, and did not drop out
   registered_attended_not_dropped =
registered_and_attended.difference(dropped_out)

   # Print the results
   print(registered_and_attended)
   print(registered_attended_not_dropped)

# Run the main function
if __name__ == "__main__":
   main()
```

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

4. 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

```
def main():
    # Read the number of tuples
    N = int(input())

    # Initialize a list to hold the inventory tuples
    inventory = []

    # Read each inventory tuple
    for _ in range(N):
        item = input().strip()
        # Remove unwanted characters and evaluate the tuple
        item_tuple = eval(item)
        inventory.append(item_tuple)

    # Read the threshold value
    threshold = int(input())

    # Initialize an empty list to hold the filtered quantities
    filtered_quantities = []

    # Loop through each item in the inventory
    for item_id, quantities in inventory:
        # Use filter to find quantities greater than the threshold
        filtered_quantities.extend(filter(lambda x: x > threshold, quantities))

    # Output the filtered quantities space-separated
    print(" ".join(map(str, filtered_quantities)))

# Run the program
if __name__ == "__main__":
    main()
```

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

5. Problem Statement

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

Write a program to help Gowshik with this task.

Example:

Given list:

[1, 2, 3, 4]

[3, 5, 2, 1]

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

Input Format

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

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

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

Output Format

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
1, 2, 3, 4
3, 5, 2, 1

Output: (4, 7, 5, 5)

Answer

```
def main():
    # Read the integer n (length of the lists)
    n = int(input())

    # Read the first list and convert it to a list of integers
    list1 = list(map(int, input().split(', ')))

    # Read the second list and convert it to a list of integers
    list2 = list(map(int, input().split(', ')))

    # Compute the element-wise sum using a list comprehension
    result = tuple(list1[i] + list2[i] for i in range(n))

    # Output the result as a tuple
    print(result)

# Run the main function
if __name__ == "__main__":
    main()
```

Status : Partially correct ***Marks : 6/10***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_MCQ

Attempt : 1
Total Mark : 20
Marks Obtained : 14

Section 1 : MCQ

1. 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***

2. 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*

3. What will be the output?

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

Answer

[‘A’, ’B’, ’C’].

Status : Correct *Marks : 1/1*

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

Answer

( )

Status : Correct *Marks : 1/1*

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

Answer

Values of a dictionary must be unique

Status : Correct *Marks : 1/1*

6. 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***

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

Output:

Tuple: (1, 3, 4)

Max value: 4

```
t=(1,)
___________
print("Tuple:" ,t)
print("Max value:",___________)
```

Answer

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

Status : Wrong ***Marks : 0/1***

8. 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

TypeError

Status : Wrong ***Marks : 0/1***

9. Which of the following is a Python tuple?

Answer

(1, 2, 3)

Status : Correct ***Marks : 1/1***

10. 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***

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

Answer

Removes an arbitrary element

Status : Correct ***Marks : 1/1***

12. Set s1 = {1, 2, 4, 3} and s2 = {1, 5, 4, 6}, find s1 &amp; s2, s1 - s2, s1 | s2 and s1 ^ s2.

Answer

s1&amp;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***

13. Which of the following isn’t true about dictionary keys?

Answer

Keys must be integers

Status : Correct ***Marks : 1/1***

14. 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***

15. What is the output of the following code?

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

Answer

Error, copy() method doesn’t exist for dictionaries

Status : Wrong ***Marks : 0/1***

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

TrueTrue

Status : Correct ***Marks : 1/1***

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

Answer

t[3] = 45

Status : Correct ***Marks : 1/1***

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

Answer

0

Status : Wrong ***Marks : 0/1***

19. What is the output of the following code?

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

Answer

True

Status : Wrong ***Marks : 0/1***

20. 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}(1, 2, 7, 8)

Status : Wrong ***Marks : 0/1***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

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

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

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

Function Signature: total_cost(item_cost)

Input Format

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

Output Format

The output consists of three lines:

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

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

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

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

Answer

```
#

# Global sales tax rate
SALES_TAX_RATE = 0.08

# Function to calculate total cost after tax
def total_cost(item_cost):
   tax_amount = item_cost * SALES_TAX_RATE
   return item_cost + tax_amount

item_cost = float(input())

final_cost = total_cost(item_cost)



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

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

2. Problem Statement

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

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

Input Format

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

The second line contains n space-separated integers.

Output Format

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
12 34 56 78 98 65 23

Output: 5
2

Answer

```
# Read input
```

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

# Print the counts
print(len(even_numbers))
print(len(odd_numbers))
```

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

3. 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 destinations(Domestic or International or Remote).

Output Format

The program outputs any one of the following:

1. If the input is valid and the destination is recognized, 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: $[calculated cost]" with two decimal places.
2. If the input weight is not a positive float, print "Invalid weight. Weight must be greater than 0."
3. If the input destination is not one of the valid options, print "Invalid destination."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5.5
Domestic

Output: Shipping cost to Domestic for a 5.5 kg package: $27.50

Answer

```
#

# Global Constants
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
    return weight * rate

# Input
weight = float(input())
destination = input()

# Compute and output
shipping_cost = calculate_shipping(weight, destination)


if shipping_cost is not None:
    print(f"Shipping cost to {destination} for a {weight} kg package: 
${shipping_cost:.2f}")
```

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

4. Problem Statement

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

Input Format

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

Output Format

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

Refer to the sample output for the exact format.

Sample Test Case

Input: 24

Output: The smallest positive divisor of 24 is: 2

Answer

```
n = int(input())

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

smallest_divisor = min(divisors[1:])

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

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_PAH_Updated

Attempt : 1
Total Mark : 60
Marks Obtained : 60

Section 1 : Coding

1. Problem Statement

Create a Python program to monitor temperatures 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 two decimal places.

Refer to the sample output for the exact format.

Sample Test Case

Input: 33.2
26.7

Output: Temperature difference: 6.50 °C

Answer

```
temp1 = float(input())
temp2 = float(input())

difference = abs(temp1 - temp2)

print(f"Temperature difference: {difference:.2f} °C")
```

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

2. Problem Statement

Ella is designing a messaging application that needs to handle long text messages efficiently. To optimize storage and transmission, she plans to implement a text compression feature that replaces consecutive repeated characters with the character followed by its count, while leaving non-repeated characters unchanged.

Help Ella create a recursive function to achieve this compression without altering the original message's meaning.

Function Specification: def compress_string(*args)

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.

Sample Test Case

Input: aaaBBBccc
Output: a3B3c3

Answer

```
def compress_string(s, index=0, count=1):
    if index == len(s) - 1:
        return s[index] + (str(count) if count > 1 else "")

    if s[index] == s[index + 1]:
        return compress_string(s, index + 1, count + 1)
    else:
        part = s[index] + (str(count) if count > 1 else "")
        return part + compress_string(s, index + 1, 1)

input_str = input()

print(compress_string(input_str))
```

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

3. Problem Statement

Sophia is developing a feature for her online banking application that calculates the total sum of digits in customers' account numbers. This sum is used to generate unique verification codes for secure transactions. She needs a program that takes an account number as input and outputs the sum of its digits.

Help Sophia to complete her task.

Function Specification: 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.

Sample Test Case

Input: 123245

Output: 17

Answer

```
num = int(input())

def sum_digits(num):
   total = 0
   for digit in str(num):
      total += int(digit)
   return total


sum = sum_digits(num)
print(sum)
```

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

4. Problem Statement

Hussain 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 BMI is a measure of a person's body fat relative to their height.

Your program should take user input for weight and height, calculate the BMI, and display the result.

Function Signature: calculate_bmi(weight, height)

Formula: BMI = Weight/(Height)2

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: [BM] followed by a float value representing the calculated BMI, rounded off two decimal points.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 70.0
1.75

Output: Your BMI is: 22.86

Answer

```
weight = float(input())
height = float(input())

def calculate_bmi(weight, height):
   bmi = weight / (height ** 2)
   print(f"Your BMI is: {bmi:.2f}")


calculate_bmi(weight, height)
```

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

5. Problem Statement

Alice works at a digital marketing company, where she analyzes large datasets. One day, she's tasked with processing customer ID numbers, which are long numeric sequences.

To simplify her task, Alice needs to calculate the digital root of each ID. The digital root is obtained by repeatedly summing the digits of a number until a single digit remains.

Help Alice write a program that reads a customer ID number, calculates its digital root, and prints the result using a loop-based approach.

For example, the sum of the digits of 98675 is 9 + 8 + 6 + 7 + 5 = 35, then 3 + 5 = 8, which is the digital root.

Function prototype: def digital_root(num)

Input Format

The input consists of an integer num.

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.

Sample Test Case

Input: 451110

Output: 3

Answer

```
num = int(input())

def digital_root(num):
    while num > 9:
        total = 0
        for digit in str(num):
            total += int(digit)
        num = total
    return num


print(digital_root(num))
```

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

6. Problem Statement

Ravi is working on analyzing a set of integers to determine how many of them are divisible by 3 and how many are divisible by 5. He decides to use lambda functions to filter 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 total 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.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 6
3 5 6 10 15 20

Output: 3
4
24
50

Answer

```
n = int(input())
numbers = list(map(int, input().split()))

div_by_3 = list(filter(lambda x: x % 3 == 0, numbers))

div_by_5 = list(filter(lambda x: x % 5 == 0, numbers))

count_3 = len(div_by_3)
sum_3 = sum(div_by_3)

count_5 = len(div_by_5)
sum_5 = sum(div_by_5)


print(count_3)
print(count_5)
print(sum_3)
print(sum_5)
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_COD_Updated

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Sneha is building a more advanced exponential calculator. She wants to implement a program that does the following:

Calculates the result of raising a given base to a specific exponent using Python’s built-in pow() function.Displays all intermediate powers from base^1 to base^exponent as a list.Calculates and displays the sum of these intermediate powers.

Help her build this program to automate her calculations.

Input Format

The input consists of line-separated two integer values representing base and exponent.

Output Format

The first line of the output prints the calculated result of raising the base to the exponent.

The second line prints a list of all powers from base^1 to base^exponent.

The third line prints the sum of all these powers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
3

Output: 8
[2, 4, 8]
14

Answer

```
base = int(input())
exponent = int(input())
result = pow(base, exponent)
powers = [pow(base, i) for i in range(1, exponent + 1)]
total = sum(powers)
print(result)
print(powers)
print(total)
```

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

2. Problem Statement

Imagine you are developing a text analysis tool for a cybersecurity company. Your task is 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 needs 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.

Sample Test Case

Input: Hello123

```
Output: Uppercase letters: 1
Lowercase letters: 4
Digits: 3
Special characters: 0
```

Answer

```
def analyze_string(input_string):


    uppercase_count = 0
    lowercase_count = 0
    digit_count = 0
    special_count = 0

    for char in input_string:
        if char.isupper():
            uppercase_count += 1
        elif char.islower():
            lowercase_count += 1
        elif char.isdigit():
            digit_count += 1
        else:
            special_count += 1

    return uppercase_count, lowercase_count, digit_count, special_count
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 : Correct ***Marks : 10/10***

3. Problem Statement

Implement a program that needs to identify Armstrong numbers. Armstrong numbers are special numbers that are equal to the sum of their digits, each raised to the power of the number of digits in the number.

Write a function is_armstrong_number(number) that checks if a given number is an Armstrong number or not.

Function Signature: armstrong_number(number)

Input Format

The first line of the input consists of a single integer, n, representing the number to be checked.

Output Format

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.

Sample Test Case

Input: 153
Output: 153 is an Armstrong number.

Answer

```
def armstrong_number(number):
   str_num = str(number)
   num_digits = len(str_num)

   total = 0
   for digit in str_num:
      total += int(digit) ** num_digits

   if total == number:
      print(f"{number} is an Armstrong number.")
   else:
      print(f"{number} is not an Armstrong number.")

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

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

4. 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.

Sample Test Case

Input: hello!!

Output: 7

Answer

```
message = input()
print(len(message))
```

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

5. Problem Statement

Sara is developing a text-processing tool that checks if a given string starts with a specific character or substring. She needs 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 Sara perform this check.

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 first line of output prints "True" if the string starts with the given character/substring, otherwise prints "False".

Refer to the sample for the formatting specifications.

Sample Test Case

Input: Examly
e

Output: False

Answer

```
main_string = input().strip()
check_substring = input().strip()

starts_with = lambda s, sub: s.startswith(sub)

print("True" if starts_with(main_string, check_substring) else "False")
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

1. What is the output of the code shown?

```
def f1():
 global x
 x+=1
 print(x)
x=12
print("x")
```

Answer

x

Status : Correct ***Marks : 1/1***

2. What is the output of the following code snippet?

```
def square(x):
    return x ** 2

result = square(4)
print(result)
```

Answer

16

Status : Correct ***Marks : 1/1***

3. What is the main advantage of using lambda functions in Python?

Answer

They allow you to write shorter code than regular functions

Status : Correct ***Marks : 1/1***

4. What keyword is used to define a lambda function in Python?

Answer

lambda

Status : Correct ***Marks : 1/1***

5. What will be the output of the following Python code?

```
def C2F(c):
    return c * 9/5 + 32
print(C2F(100))
print(C2F(0))
```

Answer

212.032.0

Status : Correct ***Marks : 1/1***

6. What is the output of the following code snippet?

```
def my_function(x):
    x += 5
    return x

a = 10
result = my_function(a)
print(a, result)
```

Answer

```
10 15
```

Status : Correct ***Marks : 1/1***

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

```
def absolute_value(x):
    if x < 0:
        return -x
    return x

result = absolute_value(-9)
print(result, absolute_value(5))
```

Answer

```
9 5
```

Status : Correct ***Marks : 1/1***

8. What is the output of the following code snippet?

```
def fun(x, y=2, z=3):
    return x + y + z

result = fun(1, z=4)
print(result)
```

Answer

7

Status : Correct ***Marks : 1/1***

9. How is a lambda function different from a regular named function in Python?

Answer

A lambda function does not have a name, while a regular function does

Status : Correct ***Marks : 1/1***

10. What is the output of the code shown?

```
def f():
 global a
 print(a)
 a = "hello"
 print(a)
a = "world"
f()
print(a)
```

Answer

worldworldhello

Status : Wrong ***Marks : 0/1***

11. What will be the output of the following Python code?

```
multiply = lambda x, y: x * y
print(multiply(2, 'Hello'))
```

Answer

HelloHello

Status : Correct ***Marks : 1/1***

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

```
num = -5
result = abs(num)
print(result)
```

Answer

5

Status : Correct ***Marks : 1/1***

13. Which of the following functions can take a lambda function as a parameter in Python?

Answer

All of the mentioned options

Status : Wrong ***Marks : 0/1***

14. What is the output of the following code?

```
x=12
def f1(a,b=x):
 print(a,b)
x=15
f1(4)
```

Answer

4 12

Status : Correct ***Marks : 1/1***

15. What will be the output of the following Python code?

```
def cube(x):
   return x * x * x
x = cube(3)
print(x)
```

Answer

27

Status : Correct *Marks : 1/1*

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Raj wants to write a program that takes a list of strings as input and returns the longest word in the list. If there are multiple words with the same length, the program should return the first one encountered.

Help Raj in his task.

Input Format

The input consists of a single line of space-separated strings.

Output Format

The output prints a string representing the longest word in the given list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: cat dog elephant lion tiger giraffe
Output: elephant

Answer

```
# Read input
words = input().split()

# Initialize variables to track the longest word
longest_word = ""
max_length = 0

# Iterate through the words
for word in words:
    if len(word) > max_length:
        longest_word = word
        max_length = len(word)

# Print the result
print(longest_word)
```

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

2. Problem Statement

You have two strings str1 and str2, both of equal length.

Write a Python program to concatenate the two strings such that the first character of str1 is followed by the first character of str2, the second character of str1 is followed by the second character of str2, and so on.

For example, if str1 is "abc" and str2 is "def", the output should be "adbecf".

Input Format

The input consists of two strings in each line.

Output Format

The output displays the concatenated string in the mentioned format.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: abc
def

Output: adbecf

Answer

```
def interleave_strings(str1, str2):
    # Check if the strings are of equal length
    if len(str1) != len(str2):
        return "Error: Strings must be of equal length"

    # Initialize an empty result string
    result = ""

    # Interleave the characters from both strings
    for i in range(len(str1)):
        result += str1[i] + str2[i]

    return result

def main():
    # Read input strings
    str1 = input().strip()
    str2 = input().strip()

    # Get the interleaved result
    result = interleave_strings(str1, str2)

    # Print the result
    print(result)

if __name__ == "__main__":
    main()
```

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

3. Problem Statement

Raja needs a program that helps him manage his shopping list efficiently. The program should allow him to perform the following operations:

Add Items: Raja should be able to add multiple items to his shopping list at once. He will input a space-separated list of items, each item being a string.

Remove Item: Raja should be able to remove a specific item from his shopping list. He will input the item he wants to remove, and if it exists in the list, it will be removed. If the item is not found, the program should notify him.

Update List: Raja might realize he forgot to add some items initially. After removing unnecessary items, he should be able to update his list by adding more items. Similar to the initial input, he will provide a space-separated list of new items.

Input Format

The first line consists of the initial list of integers should be entered as space-separated values.

The second line consists of the element to be removed should be entered as a single integer value.

The third line consists of the new elements to be appended should be entered as space-separated values.

Output Format

The output displays the current state of Raja's shopping list after each operation. After adding items, removing items, and updating the list, the program prints the updated shopping list in the following format:

"List1: [element1, element2, ... ,element_n]

List after removal: [element1, element2, ... ,element_n]

Final list: [element1, element2, ... ,element_n]".

If the item is not found in the removing item process, print the message "Element not found in the list".

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 1 2 3 4 5
3
6 7 8
Output: List1: [1, 2, 3, 4, 5]
List after removal: [1, 2, 4, 5]
Final list: [1, 2, 4, 5, 6, 7, 8]
```

Answer

```
# Step 1: Initial list input
initial_input = input().split()
list1 = initial_input.copy()  # Make a full copy of the list
print("List1:", list(map(int, list1)), end=' ')

# Step 2: Element to remove
remove_item = input()
if remove_item in list1:
    list1.remove(remove_item)
    print("List after removal:", list(map(int, list1)), end=' ')
else:
    print("Element not found in the list", end=' ')

# Step 3: New elements to append
new_items = input().split()
list1.extend(new_items)

# Final output
print("Final list:", list(map(int, list1)))
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_PAH

Attempt : 1
Total Mark : 60
Marks Obtained : 60

Section 1 : Coding

1. 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.

Sample Test Case

Input: 6
-5 2 0 -1 -10 2

Output: 1

Answer

```
# Read input
n = int(input())
arr = list(map(int, input().split()))

# Filter out only positive numbers and store in a set for O(1) lookups
positive_nums = set(x for x in arr if x > 0)

# Start checking from 1 upwards
missing = 1
while missing in positive_nums:
    missing += 1

# Output the result
print(missing)
```

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

2. Problem Statement

Gowri was doing her homework. She needed to write a paragraph about modern history. During that time, she noticed that some words were repeated repeatedly. She started counting the number of times a particular word was repeated.

Your task is to help Gowri to write a program to get a string from the user. Count the number of times a word is repeated in the string.

Note: Case-sensitive

Input Format

The first line of input consists of a string, str1.

The second line consists of a single word that needs to be counted, str2.

Output Format

The output displays the number of times the given word is in the string.

If the second string str2 is not present in the first string str1, it prints 0.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: I felt happy because I saw the others were happy and because I knew I should feel happy
happy
Output: 3

Answer

```
def count_word_occurrences(text, word):
    # Split the text into words
    # We need to handle punctuation and spaces
    import re

    # Find all occurrences of the exact word
    # Use word boundaries to ensure we match whole words
    pattern = r'\b' + re.escape(word) + r'\b'
    matches = re.findall(pattern, text)

    return len(matches)

# Get input from user
str1 = input()
str2 = input()

# Count and display result
result = count_word_occurrences(str1, str2)
print(result)
```

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

3. Problem Statement

You are tasked with writing a program that takes n integers as input from the user and stores them in a list. After this, you need to transform the list according to the following rules:

The element at index 0 should be replaced with 0.For elements at even indices (excluding index 0), replace the element with its cube.For elements at odd indices, replace the element with its square.

Additionally, you should sort the list in ascending order before applying these transformations.

Input Format

The first line of input represents the size of the list, N.

The elements of the list are represented by the next N lines.

Output Format

The first line of output displays "Original List: " followed by the original list.

The second line displays "Replaced List: " followed by the replacement list as per the given condition.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
5
1
2
3
4

Output: Original List: [1, 2, 3, 4, 5]
Replaced List: [0, 4, 27, 16, 125]

Answer

# Read input

```
N = int(input())
elements = [int(input()) for _ in range(N)]

# Sort the list
original_list = sorted(elements)

# Apply transformations
replaced_list = []
for i in range(N):
    if i == 0:
        replaced_list.append(0)
    elif i % 2 == 0:
        replaced_list.append(original_list[i] ** 3)
    else:
        replaced_list.append(original_list[i] ** 2)

# Output as per the format
print("Original List:", original_list)
print("Replaced List:", replaced_list)
```

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

4. Problem Statement

Imagine you are developing a text analysis tool for a cybersecurity company. Your task is to analyze input strings to categorize and count the characters into four categories: 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 the log entry provided as a single string.

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: {digits count}".

The fourth line contains an integer representing the count of special characters in the format "Special characters: {special characters count}".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: Hello123

Output: Uppercase letters: 1
Lowercase letters: 4
Digits: 3
Special characters: 0

Answer

```
# Read input
log_entry = input()

# Initialize counters
uppercase = sum(1 for c in log_entry if c.isupper())
lowercase = sum(1 for c in log_entry if c.islower())
digits = sum(1 for c in log_entry if c.isdigit())
special = sum(1 for c in log_entry if not c.isalnum())

# Output in the required format
print(f"Uppercase letters: {uppercase}")
print(f"Lowercase letters: {lowercase}")
print(f"Digits: {digits}")
print(f"Special characters: {special}")
```

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

5. Problem Statement

Kyara is analyzing a series of measurements taken over time. She needs to identify all the "peaks" in this list of integers.

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 single neighbor.

Your task is to find and list all such peaks using list comprehension.

Example

Input

1 3 2 4 1 5 7 6 10 2 8

Output

Peaks: [3, 4, 7, 10, 8]

Explanation

3 is a peak because it's greater than 1 and 2.

4 is a peak because it's greater than 2 and 1.

7 is a peak because it's greater than 5 and 6.

10 is a peak because it's greater than 6 and 2.

8 is a peak because it is an boundary element and it is greater than 2.

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.

Sample Test Case

Input: 1 3 2 4 1 5 7 6 10 2 8

Output: Peaks: [3, 4, 7, 10, 8]

Answer

```
# Read input
nums = list(map(int, input().split()))
n = len(nums)

# Find peaks using list comprehension
peaks = [
    nums[i] for i in range(n)
    if (i == 0 and nums[i] > nums[i+1]) or
       (i == n-1 and nums[i] > nums[i-1]) or
       (0 < i < n-1 and nums[i] > nums[i-1] and nums[i] > nums[i+1])
]

# Output the result
print("Peaks:", peaks)
```

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

6. Problem Statement

Neha is learning string operations in Python and wants to practice using built-in functions. She is given a string A, and her task is to:

Find the length of the string using a built-in function.Copy the content of A into another string B using built-in functionality.

Help Neha implement a program that efficiently performs these 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 an extra newline at the end.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: technology-23

Output: Length of the string: 13
Copied string: technology-23

Answer

```
# Read input
A = input()

# Find length using built-in function
length = len(A)

# Copy the string using built-in functionality
B = A[:]

# Print results in required format
print("Length of the string:", length, end=' ')
print("Copied string:", B)
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_COD

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Ram 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.

Ram 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 concatenated string of the first string and the reversed second string.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: hello
word

Output: hellodrow

Answer

```
# Read the two input strings
string1 = input()
string2 = input()

# Reverse the second string using slicing
reversed_string2 = string2[::-1]

# Concatenate the strings using only '+'
result = string1 + reversed_string2

# Print the result
print(result)
```

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

2. 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:

[12, 11, -13, -5, 6, -7, 5, -3, -6]

Output:

List = [-13, -5, -7, -3, -6, 12, 11, 6, 5]

Explanation:

The output is the arranged 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 sample output for the formatting specifications.

Sample Test Case

Input: [12, 11, -13, -5, 6, -7, 5, -3, -6]

Output: List = [-13, -5, -7, -3, -6, 12, 11, 6, 5]

Answer

```
# Read input
input_str = input()

# Convert the string input to a list of integers
arr = list(map(int, input_str.strip('[]').split(',')))

# Create two lists to store negatives and non-negatives separately
negatives = []
non_negatives = []

# Traverse the list and classify numbers
for num in arr:
    if num < 0:
        negatives.append(num)
    else:
        non_negatives.append(num)
```

```
# Combine the lists
result = negatives + non_negatives

# Print the output in the required format
print("List = ", result)
```

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

3. Problem Statement

Alex is working on a Python program to manage a list of elements. He needs 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 helps Alex manage the list. The program should allow Alex 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 third line of input consists of 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 of the index M.

The third line of output displays the popped element.

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 5
64
98
-1
5
26
3
Output: List after appending elements: [64, 98, -1, 5, 26]
List after popping last element: [64, 98, -1, 26]
Popped element: 5
```

Answer

```
# Read the number of elements to be appended
n = int(input())

# Initialize an empty list
elements = []

# Read and append each element to the list
for _ in range(n):
    value = int(input())
    elements.append(value)

# Read the index to pop from the list
M = int(input())

# Print the original list
print("List after appending elements:", elements)

# Pop the element at index M
popped_element = elements.pop(M)

# Print the list after popping and the popped element
print("List after popping last element:", elements)
print("Popped element:", popped_element)
```

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

4. Problem Statement

You have a string containing a phone number in the format "(XXX) XXX-XXXX". 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

(XXX) - Area code

XXX-XXXX - Phone number

Input Format

The input consists of a string, representing the phone number in the format "(XXX) XXX-XXXX".

Output Format

The output displays "Area code: " followed by a string representing the area code for the given phone number.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: (123) 456-7890
Output: Area code: 123

Answer

```
# Read the phone number string
phone_number = input()

# Extract the area code using string slicing
area_code = phone_number[1:4]

# Print the result in the required format
print("Area code:", area_code)
```

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

5. Problem Statement

Dhruv wants to write a program to slice a given string based on user-defined start and end positions.

The program should check whether the provided positions are valid and then return the sliced portion of the string if the positions are within the string's length.

Input Format

The first line consists of the input string as a string.

The second line consists of the start position (0-based index) as an integer.

The third line consists of 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.

Sample Test Case

```
Input: pythonprogramming
0
5
Output: python
```

Answer

```
# Read the input string
input_string = input()

# Read start and end positions
start = int(input())
```

```
end = int(input())

# Check if the positions are valid
if 0 <= start <= end < len(input_string):
    # Print the sliced string
    print(input_string[start:end + 1])
else:
    print("Invalid start and end positions")
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_MCQ

Attempt : 1
Total Mark : 25
Marks Obtained : 17

Section 1 : MCQ

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

```
numbers = [1, 2, 3, 4, 5]
numbers.remove(6)
print(numbers)
```

Answer

&nbsp;[1, 2, 3, 5]

Status : Wrong ***Marks : 0/1***

2. What does the following code output?

```
lst = [10, 20, 30, 40, 50]
print(lst[-4:-1])
```

Answer

[20, 30, 40]

Status : Correct ***Marks : 1/1***

3. What is the result of the slicing operation lst[-5:-2] on the list lst = [1, 2, 3, 4, 5, 6]?

Answer

[2, 3, 4]

Status : Correct ***Marks : 1/1***

4. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1]?

Answer

25

Status : Wrong ***Marks : 0/1***

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

```
numbers = [1, 2, 3, 4, 5]
numbers.append(6, 7)
print(numbers)
```

Answer

[1, 2, 3, 4, 5, (6, 7)]

Status : Wrong ***Marks : 0/1***

6. What is the output of the following Python code?

```
string1 = "Hello"
string2 = "World"
result = string1 + string2
print(result)
```

Answer

Hello World

Status : Wrong ***Marks : 0/1***

7. Which of the following is a valid way to use the '%' operator to concatenate strings in Python?

Answer

string1 * %s * string2

Status : Wrong ***Marks : 0/1***

8. What is the output of the following Python code?

```
a = "Hello"
b = "World"
c = a + " " + b
print(c)
```

Answer

Hello World

Status : Correct ***Marks : 1/1***

9. What is the output of the following Python code?

```
b = "Projects!"
print(b[2:5])
```

Answer

oje

Status : Correct ***Marks : 1/1***

10. Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], Which of the following is the correct syntax for slicing operation?

Answer

print(list1[:-2])

Status : Wrong ***Marks : 0/1***

11. Which method is used to add multiple items to the end of a list?

Answer

extend()

Status : Correct ***Marks : 1/1***

12. What is the output of the following Python code?

```
text = "Python"
result = text.center(10, "*")
print(result)
```

Answer

Python

Status : Correct ***Marks : 1/1***

13. What is the output of the following Python code?

```
word = "programming"
answer = word.index("gram")
print(answer)
```

Answer

3

Status : Correct ***Marks : 1/1***

14. Which method in Python is used to create an empty list?

Answer

list()

Status : Correct ***Marks : 1/1***

15. What is the output of the following code?

```
my_list = [1, 2, 3]
my_list *= 2
print(len(my_list))
```

Answer

3

Status : Wrong ***Marks : 0/1***

16. What is the output of the following Python code?

```
text = "   Python   "
answer = text.strip()
print(answer)
```

Answer

Python

Status : Correct ***Marks : 1/1***

17. What is the output of the following Python code?

```
name = "John"
age = 25
message = "My name is %s and I am %d years old." % (name, age)
print(message)
```

Answer

My name is John and I am 25 years old.

Status : Correct ***Marks : 1/1***

18. If you have a list lst = [1, 2, 3, 4, 5, 6], what does the slicing operation

lst[-3:] return?

Answer

The last three elements of the list

Status : Correct ***Marks : 1/1***

19. What is the output of the following code?

```
my_list = [3, 6, 1, 2, 5, 4]
print(sorted(my_list) == my_list.sort())
```

Answer

False

Status : Correct ***Marks : 1/1***

20. What is the output of the following Python code?

```
txt = "My Classroom"
print(txt.find("o"))
print(txt.index("o"))
```

Answer

99

Status : Correct ***Marks : 1/1***

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

```
my_list = [1, 2, 2, 3]
print(my_list.count(2))
```

Answer

2

Status : Correct ***Marks : 1/1***

22. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1]?

Answer

25

Status : Correct ***Marks : 1/1***

23. What is the output of the following Python code?

```
word = "Python"
result = word[::-1]
print(result)
```

Answer

onhtyP

Status : Wrong ***Marks : 0/1***

24. What does the append() method do in Python?

Answer

Adds a new element to the end of the list

Status : Correct ***Marks : 1/1***

25. What does negative indexing in Python lists allow you to do?

Answer

Access elements in the list from the end

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_PAH_Updated

Attempt : 1
Total Mark : 60
Marks Obtained : 60

Section 1 : Coding

1. Problem Statement

Aarav is fascinated by the concept of summing numbers separately based on their properties. He plans to write a program that calculates the sum of even numbers and odd numbers separately from 1 to a given positive integer.

Aarav wants to input an integer value to represent the upper limit of the range. Help Aarav 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 N 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.

Sample Test Case

Input: 10

Output: Sum of even numbers from 1 to 10 is 30
Sum of odd numbers from 1 to 10 is 25

Answer

```
# You are using Python
a=int(input())
e=0
o=0
for i in range(1,a+1):
   if(i%2==0):
      e+=i
   else:
         o+=i
print("Sum of even numbers from 1 to",a,"is",e)
print("Sum of odd numbers from 1 to",a,"is",o)
```

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

2. Problem Statement

Sophia, 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 Sophia compute the sum of all numbers between start and end (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 3.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
10

Output: 37

Answer

```
# You are using Python
a=int(input())
b=int(input())
s=0
for i in range(a,b+1):
   if(i%3!=0):
      s+=i
print(s)
```

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

3. Problem Statement

As a software engineer, your goal is 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 years 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 lines.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 2000
2053
```

```
Output: 2000
2004
2008
2012
2016
2020
2024
2028
2032
2036
2040
2044
2048
2052
```

Answer

```
# You are using Python
a=int(input())
b=int(input())
for i in range(a,b+1):
    if(i%4==0 and i%100!=0):
        print(i)
    elif(i%400==0):
        print(i)
```

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

4. Problem Statement

Imagine being entrusted with the responsibility of creating a program that simulates a math workshop for students. Your task is to develop an interactive program that not only calculates but also showcases the charm of factorial values.Your program should efficiently compute and present the sum of digits for factorial values of only odd numbers within a designated range. This approach will ingeniously keep even factorials at bay, allowing students to delve into the intriguing world of mathematics with enthusiasm and clarity.

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.

Sample Test Case

Input: 6

```
Output: 1! = 1, sum of digits = 1
3! = 6, sum of digits = 6
5! = 120, sum of digits = 3
```

Answer

```
# You are using Python
import math
def digits(n):
   return sum(int(digit) for digit in str(n))
a=int(input())
print("1! = 1,sum of digits = 1")
for i in range(3,a+1,2):
   fact=math.factorial(i)
   som=digits(fact)
   print(i,"! =",fact,", sum of digits =",som)
```

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

5. Problem Statement

Rajesh 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 accordingly. If the process exceeds 100 steps, the program should print a message indicating so and use break to exit.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the total number of steps taken to reach 1 if it's under 100.

If it's more than 100, it displays "Exceeded 100 steps. Exiting...".

Refer to sample output for the formatting specifications.

Sample Test Case

Input: 6

Output: Steps taken to reach 1: 8

Answer

```
# You are using Python
a=int(input())
s=0
```

```
while(a!=1):
   if(a%2==0):
      a=a//2
      s+=1
   else:
      a=(a*3)+1
      s+=1
if(s>100):
   print("Exceeded 100 steps. Exiting...")
else:
   print("Steps taken to reach 1:",s)
```

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

6. Problem Statement

Kamali recently received her electricity bill and wants to calculate the amount she needs to pay based on her usage. The electricity company charges 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 Kamali calculate the amount she needs to pay for her 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 Kamali needs to pay in the format "Rs. amount".

Refer to the sample output for the exact format.

Sample Test Case

Input: 350

Output: Rs. 2000

Answer

```
# You are using Python
a=int(input())
b=0
if a>200:
    b+=(a-200)*10
    a=200
if a>100:
    b+=(a-100)*5
print(f"Rs. {b}")
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Students are allowed to work on our computer center machines only 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: Here, secret code means 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.

Sample Test Case

Input: 2345

Output: Incorrect code

Answer

```
# Read the secret code as an integer
n = int(input().strip())

# Convert the number to a string to extract digits easily
n_str = str(n)

# Extract the first digit and the last three digits
first_digit = int(n_str[0])
last_three_digits = int(n_str[-3:])

# Check the condition: last three digits should be divisible by the first digit
if last_three_digits % first_digit == 0:
   print("Logged In")
else:
   print("Incorrect code")
```

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

2. Problem Statement

Gabriel is working on a wildlife research project where he needs to compute various metrics for different animals based on their characteristics. Each animal type requires a different calculation: a deer's distance traveled, a bear's weight based on footprint size, or a bird's altitude based on its flying pattern.

Conditions:

For Deer (Mode 'D' or 'd'): Distance = speed of sound * time taken, where the speed of sound in air is 343 meters per second.For Bear (Mode 'B' or 'b'): Weight = footprint size * average weight, where the average weight per square inch for a bear is 5.0 pounds.For Bird (Mode 'F' or 'f'): Altitude = flying pattern * distance covered (in meters).

Write a program to help Gabriel analyze the characteristics of animals based on the given inputs.

Input Format

The first line of input consists of a character, representing the type of animal 'D/d' for deer, 'B/b' for bear, and 'F/f' for bird.

If the choice is 'D' or 'd':

The second line of input consists of a floating-point value T, representing the time taken from the deer's location to the observer.

If the choice is 'B' or 'b':

The second line of input consists of a floating-point value S, representing the size of the bear's footprint in square inches.

If the choice is 'F' or 'f':

1. The second line of input consists of a floating-point value P, representing the bird's flying pattern.
2. The third line consists of a floating-point value D, representing the distance covered by the bird in meters.

Output Format

The output prints one of the following:

If the choice is 'D' or 'd':

The output prints "Distance: X m" where X is a floating point value rounded off to two decimal places, representing the calculated distance traveled by the sound wave in meters.

If the choice is 'B' or 'b':

The output prints "Weight: Y lb" where Y is a floating point value rounded off to two decimal places, representing the estimated weight of the bear in pounds.

If the choice is 'F' or 'f':

The output prints "Altitude: Z m" where Z is a floating point value rounded off to two decimal places, representing the calculated altitude of the bird's flight in meters.

If the given choice is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: d
2.5

Output: Distance: 857.50 m

Answer

```
# Read the type of animal
animal_type = input().strip()

# Constants
SPEED_OF_SOUND = 343  # meters per second
BEAR_WEIGHT_FACTOR = 5.0  # pounds per square inch

# Processing based on animal type
if animal_type in ('D', 'd'):
    T = float(input().strip())
    if 1.0 <= T <= 10.0:
        distance = SPEED_OF_SOUND * T
        print(f"Distance: {distance:.2f} m")
    else:
        print("Invalid")

elif animal_type in ('B', 'b'):
    S = float(input().strip())
    if 1.0 <= S <= 5.0:
        weight = S * BEAR_WEIGHT_FACTOR
        print(f"Weight: {weight:.2f} lb")
    else:
        print("Invalid")

elif animal_type in ('F', 'f'):
    P = float(input().strip())
    D = float(input().strip())
    if 1.0 <= P <= 50.0 and 1.0 <= D <= 50.0:
        altitude = P * D
        print(f"Altitude: {altitude:.2f} m")
    else:
        print("Invalid")

else:
    print("Invalid")
```

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

3. Problem Statement

Alex is practicing programming and is curious about prime and non-prime digits. He wants to write a program that calculates the sum of the non-prime digits in a given integer using loops.

Help Alex to complete his task.

Example:

Input:

845

output:

12

Explanation:

Digits: 8 (non-prime), 4 (non-prime), 5 (prime)

The sum of Non-Prime Digits: 8 + 4 = 12

Output: 12

Input Format

The input consists of a single integer X.

Output Format

The output prints an integer representing the sum of non-prime digits in X.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 845

Output: 12

Answer

```
# Function to check if a digit is prime
def is_prime(digit):
   return digit in {2, 3, 5, 7}  # Only 2, 3, 5, and 7 are prime single-digit numbers

# Read input number
X = int(input().strip())

# Initialize sum of non-prime digits
sum_non_prime = 0

# Iterate over each digit in the number
for digit in str(X):
   num = int(digit)
   if not is_prime(num):  # If digit is not prime, add to sum
      sum_non_prime += num

# Print the result
print(sum_non_prime)
```

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

4. Problem Statement

Max is fascinated by prime numbers and the Fibonacci sequence. He wants to combine these two interests by creating a program that outputs the first n prime numbers within the Fibonacci sequence.

Your task is to help Max by writing a program that prints the first n prime numbers in the Fibonacci sequence using a while loop along with the break statement to achieve the desired functionality.

Input Format

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

Output Format

The output displays space-separated first n prime numbers found in the Fibonacci sequence.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

Output: 2 3 5 13 89

Answer

```
# Function to check if a number is prime
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

# Read input
n = int(input().strip())
```

```
# Initialize Fibonacci sequence
a, b = 0, 1
prime_fib_numbers = []  # List to store prime Fibonacci numbers

# Generate Fibonacci sequence and check for prime numbers
while len(prime_fib_numbers) < n:
    if is_prime(b):
        prime_fib_numbers.append(b)
    a, b = b, a + b  # Move to the next Fibonacci number

# Print space-separated output
print(" ".join(map(str, prime_fib_numbers)))
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_COD_Updated

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. 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

The output displays space-separated consonants present in the sentence.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: Hello World!
Output: H l l W r l d

Answer

```
# You are using Python
sen=input()
vow='aeiouAEIOU'
con=[ch for ch in sen if ch.isalpha() and ch not in vow]
print(''.join(con))
```

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

2. Problem Statement

You work as an instructor at a math enrichment program, and your goal is to develop a program that showcases the concept of using control statements to manipulate 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 represents the upper limit of the range.

Output Format

The output displays the square of even numbers from 1 to 'n' separated by lines.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10
Output: 4
16
36
64
100

Answer

```
n=int(input())
for i in range(1,n+1):
    if i%2!=0:
        continue
    print(i**2)
```

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

3. 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 excludes all numbers that are palindromes (numbers that read the same backward as forward).

Help John find the total sum of the digits of non-palindromic numbers in the range [start, end] (both inclusive).

Example:

Input:

10

20

Output:

55

Explanation:

Range [10, 20]: Non-palindromic numbers are 10, 12, 13, 14, 15, 16, 17, 18, 19 and 20.

Digit sums: 1+0 + 1+2 + 1+3 + 1+4 + 1+5 + 1+6 + 1+7 + 1+8 + 1+9 + 2+0 = 55.

Output: 55

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 non-palindromic numbers in the range.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10
20

Output: 55

Answer

```
def ispal(n):
   return str(n)==str(n)[::-1]

def sumofd(num):
   return sum(int(digit) for digit in str(num))

start=int(input())
end=int(input())
tot=sum(sumofd(i) for i in range(start,end+1) if not ispal(i))
print(tot)
```

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

4. Problem Statement

Ethan, a curious mathematician, is fascinated by perfect numbers. A perfect number is a number that equals the sum of its proper divisors (excluding itself). Ethan wants to identify all perfect numbers within a given range.

Help him 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 a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
100

Output: 6 28

Answer

```
# You are using Python
def isper(num):
   return sum(i for i in range(1,num) if num % i ==0)==num

start=int(input())
end=int(input())
per=[i for i in range(start,end+1) if isper(i)]
print(*per)
```

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

5. Problem Statement

Emma, a mathematics enthusiast, is exploring a range of numbers and wants to count how many of them are not Fibonacci numbers.

Help Emma determine the count of non-Fibonacci numbers within the 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.

Sample Test Case

Input: 1
10

Output: 5

Answer

```
def isfib(n):
   a,b=0,1
   while b<n:
      a,b=b,a+b
   return b==n

def countnon(start,end):
   count=0
   for i in range(start,end+1):
      if(isfib(i)):
         continue
```

```
        count+=1
    return count

start=int(input())
end=int(input())
print(countnon(start,end))
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 9

Section 1 : MCQ

1. What will be the output of the following Python code?

```
i = 1
while True:
   if i % 2 == 0:
      i += 1
      continue
   if i > 10:
      break
   print(i, end = " ")
   i += 2
```

Answer

1 3 5 7 9

Status : Correct ***Marks : 1/1***

2. What is the output of the following?

```
i=0
while(1):
  i++
  print i
  if(i==4):
    break
```

Answer

Syntax Error

Status : Correct ***Marks : 1/1***

3. What is the output of the following code?

```
i = 5
while True:
    if i%0O9 == 0:
        break
    print(i)
    i += 1
```

Answer

5 6 7 8

Status : Wrong ***Marks : 0/1***

4. Which keyword is used to immediately terminate a loop?

Answer

break

Status : Correct ***Marks : 1/1***

5. What is the output of the following?

```
for i in range(10):
    if i == 5:
        break
    else:
        print(i, end=' ')
else:
    print("Here")
```

Answer

0 1 2 3 4 Here

Status : Wrong ***Marks : 0/1***

6. What will the following code output?

```
x = 0
while x < 5:
    if x == 3:
        break
    x += 1
else:
    print("Completed")
print(x)
```

Answer

3

Status : Correct ***Marks : 1/1***

7. What is the output of the following program?

```
i=1
while(i<3):
  j=0
  while(j<3):
    print(i%3,end=" ")
    j=j+1
  i=i+1
```

Answer

0 0 0 1 1 1

Status : Wrong ***Marks : 0/1***

8. Which keyword used in loops can skip the remaining statements for a particular iteration and start the next iteration?

Answer

continue

Status : Correct ***Marks : 1/1***

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

```
i = 1
while True:
   if i%3 == 0:
      break
   print(i)
   i += 1
```

Answer

1 2

Status : Correct ***Marks : 1/1***

10. What will be the output of the following code snippet?

```
i = 0
while i < 5:
   if i % 2 == 0:
      i += 1
      continue
   print(i, end=" ")
   i += 1
```

Answer

1 3

Status : Correct ***Marks : 1/1***

11. What will be the output of the following Python code?

```
i = 5
while True:
   if i%0O11 == 0:
      break
   print(i, end = " ")
   i += 1
```

Answer

5 6 7 8 9 10

Status : Wrong ***Marks : 0/1***

12. When does the else statement written after the loop execute?

Answer

When loop condition becomes false

Status : Correct ***Marks : 1/1***

13. What will be the output of the following Python code?

```
i = 1
while True:
   if i%3 == 0:
      break
   print(i)
   i + = 1
```

Answer

1 2

Status : Wrong ***Marks : 0/1***

14. What will be the output of the following Python code?

```
i = 1
while False:
    if i%2 == 0:
        break
    print(i)
    i += 2
```

Answer

The code runs successfully but does not print anything

Status : Correct ***Marks : 1/1***

15. How many times will the inner for loop be executed in the below code?

```
i=0
while(True):
   for j in range(4,0,-2):
     print(i*j)
     print('')
     i=i+1
   if(i%2==0):
     break
```

Answer

2

Status : Wrong ***Marks : 0/1***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_PAH

Attempt : 1
Total Mark : 6
Marks Obtained : 4.7

Section 1 : Coding

1. Problem Statement

Shawn, a passionate baker, is planning to bake cookies for a large party. His original recipe makes 15 cookies, with the following ingredient quantities: 2.5 cups of flour, 1 cup of sugar, and 0.5 cups of butter.

Write a program to calculate the amounts of flour, sugar, and butter needed for a different number of cookies. Provide the ingredient quantities for a specified number of cookies, maintaining the original proportions 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 for n cookies, as a double value rounded to two decimal places.

The second line prints "Sugar: Y cups" where Y represents the amount of Sugar required for n, as a double value rounded to two decimal places.

The third line prints "Butter: Z cups" where Z represents the amount of flour required for n, as a double value rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 15

Output: Flour: 2.50 cups
Sugar: 1.00 cups
Butter: 0.50 cups

Answer

```
# You are using Python
n=int(input())
f,s,b=(n/15)*2.5,(n/15)*1,(n/15)*0.5
print(f"Flour: {f:.2f} cups\nSugar: {s:.2f} cups\nButter:{b:.2f} cups")
```

Status : Correct ***Marks : 1/1***

2. 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).

Display the room's climate details.

Input Format

The first line of input consists of a string, representing the room name.

The second line consists of a float value, representing the temperature.

The third line consists of 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.

Sample Test Case

Input: Living Room
23.45
45.78

Output: Room: Living Room
Temperature: 23.45
Humidity: 45.78%

Answer

```
# You are using Python
a=str(input())
b=float(input())
c=float(input())
print("Room:",a)
print("Temperature:",b)
print("Humidity:",c,"%")
```

Status : Partially correct ***Marks : 0.3/1***

3. Problem Statement

Ella, an avid TV show enthusiast, is planning a binge-watching marathon for a new series. She has a specific routine: after watching a set number of episodes, she takes a short break.

She is provided with the following information:

Each episode of the series has a fixed duration of 45 minutes.After a certain number of episodes, there is a break of 15 minutes.

Ella wants to know the total time she will need to watch the entire series, including the breaks. Your task is to help Ella by calculating the total viewing time.

Input Format

The first line of input consists of an integer E, representing the total number of episodes in the series.

The second line consists of an integer B, 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.

Sample Test Case

Input: 5
2

Output: 255 minutes

Answer

```
# You are using Python
e=int(input())
b=int(input())
rqtime=(e*45)+((e-1)//b)*15
print(f"{rqtime} minutes")
```

Status : Correct *Marks : 1/1*

4. Problem Statement

Oliver is planning a 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 in time, Oliver needs to calculate the download time.

Your 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 MB * 8 bits/byte = 6400 megabits) and divide it by the download speed (6400 Mbps / 40 Mbps = 160 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

The output prints "Download Time: X hours, Y minutes, and Z seconds", where X, Y, and Z are integers representing the hours, minutes, and seconds respectively.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 180
3
Output: Download Time: 0 hours, 8 minutes, and 0 seconds

Answer

```
# You are using Python
n=int(input())
s=int(input())
n=n*8
speed=n/s
h,m,s=speed//3600,(speed%3600)//60,speed%60
print(f"Download Time: {int(h)} hours, {int(m)} minutes, and {int(s)} seconds")
```

Status : Correct ***Marks : 1/1***

5. Problem Statement

Liam works at a car dealership and is responsible for recording the details of cars that arrive at 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 summary.

Assist him in 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 Make: ", 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.

Sample Test Case

Input: Toyota
Camry
23450.75

Output: Car Make: Toyota
Car Model: Camry
Price: Rs.23450.75

Answer

```
# You are using Python
a=str(input())
b=str(input())
c=float(input())
print("Car Make: ",a)
print("Car Model: ",b)
print("Price: Rs.",c)
```

Status : Partially correct ***Marks : 0.4/1***

6. 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 and its triple to the integer itself is always 4. Mandy, intrigued by this statement, decides to validate it using logical operators and basic arithmetic.

She wants to confirm if the statement holds true for any positive integer n.

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 "Rachel's statement is: " followed by a Boolean value indicating whether Rachel's statement is correct.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 12

Output: Sum: 48
Rachel's statement is: True

Answer

```
# You are using Python
a=int(input())
sum=a*4
print("Sum: ",sum)
print("Rachel's statement is: True")
```

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Mandy is working on a mathematical research project involving complex numbers. For her calculations, she often needs to swap the real and imaginary parts of two complex numbers.

Mandy needs a Python program that takes two complex numbers as input and swaps their real and imaginary values.

Input Format

The first line of input consists of a complex number in the format a+bj, representing the first complex number.

The second line consists of a complex number in the format a+bj, representing the second complex number.

Output Format

The first line of output displays "New first complex number: " followed by the swapped complex number.

The second line of output displays "New second complex number: " followed by the swapped complex number.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10+8j
7-9j

Output: New first complex number: (8+10j)
New second complex number: (-9+7j)

Answer

```
# You are usia=comle
a=complex(input())
b=complex(input())
a1=complex(a.imag,a.real)
b1=complex(b.imag,b.real)
print("New first complex number: ",a1)
print("New second complex number: ",b1)
```

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

2. Problem Statement

Olivia is creating a wellness dashboard for her new fitness app, FitTrack. She needs a program that can capture and display key details about a user's workout. The program should read the user's full name, the total steps they ran, the energy they expended in kilojoules, and the duration of their workout in hours. After collecting this information, the program will generate a detailed summary of the user's fitness activity.

Your task is to guide Olivia 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: X hours" where X is the workout duration, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: Alex
10000
350.5
1.5
```

```
Output: User Name: Alex
Total Steps: 10000
Calories Burned: 350.5
Workout Duration: 1.5 hours
```

Answer

```
# You are using Python
a=str(input())
b=int(input())
c=float(input())
```

```
d=float(input())
print("User Name: ",a)
print("Total Steps: ",b)
print("Calories Burned: ",c)
print("Workout Duration: ",d," hours")
```

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

3. Problem Statement

John is developing a financial application to help users manage their investment portfolios. As part of the application, he needs to write a program that receives the portfolio's main value and the values of two specific investments as inputs. The program should then display these values in reverse order for clear visualization.

Help John achieve this functionality by writing the required program.

Input Format

The first line of input consists of a float, representing the first investment value.

The second line of input consists of a float, representing the second investment value.

The third line of input consists of an integer, representing the portfolio ID.

Output Format

The first line of output prints "The values in the reverse order:".

The second line prints the integer, representing the portfolio ID.

The third line prints the second float, representing the second investment value.

The fourth line prints the first float, representing the first investment value.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 35.29
9374.11
48

Output: The values in the reverse order:
48
9374.11
35.29

Answer

```
# You are using Python
a=float(input())
b=float(input())
c=int(input())
print("The values in the reverse order:")
print("\n",c)
print("\n",b)
print("\n",a)
```

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

4. Problem Statement

Liam and his friends are sharing the cost of a group purchase. The total cost of the purchase is subject to a 10% discount. One of the friends receives a 35% bonus, which means they will pay a larger portion of the discounted cost. The remaining cost is then divided equally among the other friends.

Write a program to:

Calculate the total cost after applying a 10% discount.Determine the amount paid by the friend who receives a 35% bonus.Calculate the amount each of the other friends will pay.

Input Format

The first line of input consists of a float value f, representing the total cost.

The second line contains an integer value n, representing the total number of friends.

Output Format

The first line of output displays "Cost after a 10% discount: " followed by the discounted cost of the ticket package as a float value formatted to two decimal places.

The second line displays "Friend with a 35% bonus pays: " followed by the amount paid by the friend with the bonus as a float value formatted to two decimal places.

The third line displays "Each of the other friends pays: " followed by the individual share of the remaining cost as a float value formatted to two decimal places.

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 10000.0
5
Output: Cost after a 10% discount: 9000.00
Friend with a 35% bonus pays: 3150.00
Each of the other friends pays: 1462.50
```

Answer

```
# You are using Python
a=float(input())
b=int(input())
c=a*(90/100)
d=c*(35/100)
e=(c-d)/(b-1)
print(f"Cost after a 10% discount: {c:.2f} ")
print(f"Friend with a 35% bonus pays: {d:.2f}")
print(f"Each of the other friends pays: {e:.2f}")
```

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

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_COD

Attempt : 1
Total Mark : 5
Marks Obtained : 4

Section 1 : Coding

1. 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 N, representing Alice's salary.

The second line consists of a float value F, representing Bob's salary.

Output Format

The first line of output displays "Initial salaries:"

The second line displays "Alice's salary = N", where N is Alice's salary.

The third line of output displays "Bob's salary = F", where F is Bob's salary.

After a new line space, the following line displays "New salaries after swapping:"

The next line displays "Alice's salary = X", where X is the swapped salary.

The last line displays "Bob's salary = Y", where Y is the swapped salary.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10000
15400.55

Output: Initial salaries:
Alice's salary = 10000
Bob's salary = 15400.55

New salaries after swapping:
Alice's salary = 15400.55
Bob's salary = 10000

Answer

```
# You are using Python
a=int(input())
b=float(input())
print("Initial salaries:")
print("Alice's salary = ",a)
print("Bob's salary = ",b)
a,b=b,a
print("New salaries after swapping:")
print("Alice's salary = ",a)
print("Bob's salary = ",b)
```

Status : Correct ***Marks : 1/1***

2. Problem Statement

Quentin, a mathematics enthusiast, is exploring the properties of numbers. He believes that for any set of four consecutive integers, calculating the average of their fourth powers and then subtracting the product of the first and last numbers yields a constant value.

To validate his hypothesis, check if the result is indeed constant and display.

Example:

Input:

5

Output:

Constant value: 2064.5

Explanation:

Find the Average:

Average: (625 + 1296 + 2401 + 4096)/4 = 2104.5

Now, we calculate the product of a and (a + 3):

Product = 5 × (5 + 3) = 5 × 8 = 40

Final result: 2104.5 - 40 = 2064.5

Input Format

The input consists of an integer a, representing the first of four consecutive integers.

Output Format

The output displays "Constant value: " followed by the computed result based on Quentin's formula.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Output: Constant value: 2064.5

Answer

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

sum=a**4+(a+1)**4+(a+2)**4+(a+3)**4
average=sum/4
c=a*(a+3)
d=average-c
print("Constant value: ",d)
```

Status : Correct ***Marks : 1/1***

3. 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, F.

Output Format

The output prints "The integer value of F is: {result}", followed by the integer number equivalent to the floating point number.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10.36

Output: The integer value of 10.36 is: 10

Answer

```
# You are using Python
a=float(input())
b=int(a)
print("The integer value of ",a,"is: ",b)
```

Status : Correct ***Marks : 1/1***

4. 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 for each task, with a base allowance of Rs.50. The younger child receives an allowance rate of Rs.3 for each task, with a base allowance of Rs.30.

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 allowance for both children combined.

Input Format

The first line of input consists of an integer n, representing the number of chores completed by the older child.

The second line consists of an integer m, representing the number of chores completed by the youngest child.

Output Format

The first line of output displays "Older child allowance: Rs." followed by an integer representing the allowance calculated for the older sibling.

The second line displays "Younger child allowance: Rs." followed by an integer representing the allowance calculated for the youngest sibling.

The third line displays "Total allowance: Rs." followed by an integer representing the sum of both siblings' allowances.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10
5
Output: Older child allowance: Rs.100
Younger child allowance: Rs.45
Total allowance: Rs.145

Answer

```
n=int(input())
m=int(input())
a=(n*3)+50
b=(m*3)+30
print("Older child allowance: Rs.",a)
print("Younger child allowance: Rs.",b)
print("Total allowance: Rs.",a+b)
```

Status : Wrong ***Marks : 0/1***

5. Problem Statement

Bob, the owner of a popular bakery, wants to create a special offer code for his customers. To generate the code, he plans to combine the day of the month with the number of items left in stock.

Help Bob to encode these two values into a unique offer code.

Note: Use the bitwise operator to calculate the offer code.

Example

Input:

15

9

Output:

Offer code: 6

Explanation:

Given the day of the month 15th day (binary 1111) and there are 9 items left (binary 1001), the offer code is calculated as 0110 which is 6.

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 S, representing the number of items left in stock.

Output Format

The output displays "Offer code: " followed by an integer representing the encoded offer code.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 15
9

Output: Offer code: 6

Answer

```
# You are using Python
a=int(input())
b=int(input())
c=a^b
print("Offer code: ",c)
```

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: Saravanan .G
Email: 241501189@rajalakshmi.edu.in
Roll no:
Phone: 6382117108
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 14

Section 1 : MCQ

1. Which of the following expressions results in an error?

Answer

int('10')

Status : Wrong *Marks : 0/1*

2. What is the value of the following expression?

float(22//3+3/3)

Answer

8.0

Status : Correct *Marks : 1/1*

3. Which of the following functions converts a string to a float in Python?

Answer

float(x)

Status : Correct ***Marks : 1/1***

4. Which of the following represents the bitwise XOR operator?

Answer

^

Status : Correct ***Marks : 1/1***

5. What is typecasting in Python?

Answer

Change data type property

Status : Correct ***Marks : 1/1***

6. What is the output of the following number conversion?

```
z = complex(1.25)
print(z)
```

Answer

(1.25+0j)

Status : Correct ***Marks : 1/1***

7. What does 3 ^ 4 evaluate to?

Answer

7

Status : Correct ***Marks : 1/1***

8. What is used to concatenate two strings in Python?

Answer

+ operator

Status : Correct ***Marks : 1/1***

9. What is the value of the following expression?

8/4/2, 8/(4/2)

Answer

(1.0,4.0)

Status : Correct ***Marks : 1/1***

10. Which of the following operators has its associativity from right to left?

Answer

**

Status : Correct ***Marks : 1/1***

11. What is the value of x in the following program?

```
x = int(43.55+2/2)
print(x)
```

Answer

44

Status : Correct ***Marks : 1/1***

12. What is the output of the below expression?

```
print(3*1**3)
```

Answer

3

Status : Correct ***Marks : 1/1***

13. Evaluate the expression given below if A= 16 and B = 15

A % B // A

Answer

0

Status : Correct ***Marks : 1/1***

14. What is the return type of the function id?

Answer

int

Status : Correct ***Marks : 1/1***

15. What is the output of the following program?

```
print((1, 2) + (3, 4))
```

Answer

```
(1, 2, 3, 4)
```

Status : Correct ***Marks : 1/1***