Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

1. Which of these is not a core data type?

Answer

Class

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

2. Which is the correct operator for power(xy)?

Answer

x**y

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

3. What is the output of the below expression?

print(3*1**3)

Answer

3

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

4. Which of the following can convert the string to a float number?

Answer

float(str)

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

5. Evaluate the expression given below if A= 16 and B = 15

A % B // A

Answer

0.0

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

6. What does 3 ^ 4 evaluate to?

Answer

7

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

7. Which of the following is an example of the type casting?

Answer

All of the above

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

8. What is the output of the following program?

```
print((1, 2) + (3, 4))
```

Answer

(1, 2, 3, 4)

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

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

```
x = int(34.56 - 2 * 2)
print(x)
```

Answer

30

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

10. What will be the output for the below code?

```
x=15
y=12
print(x&y)
```

Answer

12

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

11. What is used to concatenate two strings in Python?

Answer

+ operator

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

12. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same.

True or False?

Answer

False

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

13. Which of the following operators has its associativity from right to left?

Answer

**

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

14. 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***

15. What is the output of the following number conversion?

```
z = complex(1.25)
print(z)
```

Answer

(1.25+0j)

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_COD

Attempt : 1
Total Mark : 5
Marks Obtained : 5

Section 1 : Coding

1. 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

```
a=int(input())
a1=a**4
a2=(a+1)**4
a3=(a+2)**4
a4=(a+3)**4
avg=(a1+a2+a3+a4)/4
prd=a*(a+3)
c=(avg)-prd
print(F"Constant value: {c:.1f}")
```

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

2. 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())
print("The integer value of",a,"is:",int(a))
```

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

3. 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

```
al_sal=int(input())
bob_sal=float(input())
print("Initial salaries:")
print("Alice's salary =",al_sal)
print("Bob's salary =",bob_sal)
al_sal,bob_sal=bob_sal,al_sal
print("\nNew salaries after swapping:")
print("Alice's salary =",al_sal)
print("Bob's salary =",bob_sal)
```

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

4. 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

```
a=int(input())
b=int(input())
print("Offer code: ",a^b)
```

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

5. 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

```
# You are using Python
n=int(input())
m=int(input())
ol=50+(5*n)
nh=30+(3*m)
tot=ol+nh
print("Older child allowance:Rs.",ol)
print("Younger child allowance:Rs.",nh)
print ("Total allowance:Rs.",tot)
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 11

Section 1 : MCQ

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

```
i = 1
while True:
   if i%0O7 == 0:
      break
   print(i)
   i += 1
```

Answer

1 2 3 4 5 6

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

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

Answer

break

Status : Correct *Marks : 1/1*

3. 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*

4. 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*

5. What is the output of the following code?

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

Answer

0 1 2 3 4 Here

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

6. 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)
    i += 2
```

Answer

1 3 5 7 9

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

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

```
i = 0
for i in range(10):
    break
print(i)
```

Answer

0

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

8. What is the output of the following?

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

Answer

1 2 3 4

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

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

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. 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***

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

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

Answer

012

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

13. What is the output of the following?

```
i = 2
while True:
  if i%3 == 0:
    break
  print(i)
  i += 2
```

Answer

2 4

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

14. 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***

15. 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***

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

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

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

```
start = int(input())
end = int(input())
fibonacci = set()
a ,b = 0, 1
while a<= 100:
   fibonacci.add(a)
   a, b = b, a + b

count = 0
for i in range(start, end +1):
   if i in fibonacci:
      continue
   count +=1
print(count)
```

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

2. 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
sentence = input()
result =  ""
for char in sentence:
   if not (char.isalpha()):
      continue
   lower_char = char.lower()
   if lower_char in ['a', 'e', 'i', 'o', 'u']:
      continue
   result += char + " "
print(result.strip())
```

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

3. 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
start = int(input())
end = int(input())

perfect_numbers = [ ]

for num in range(start, end + 1):
   sum_divisors = 0

   for i in range(1, num):
      if num % i== 0:
         sum_divisors += i

   if sum_divisors == num:

       perfect_numbers.append(str(num))

print(" ".join(perfect_numbers))
```

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

4. 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

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

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

Status : Correct *Marks : 10/10*

5. 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

```
# You are using Python
start = int(input())
end = int(input())

total_sum = 0

for num in range (start, end + 1):
    if str(num) == str(num)[::-1]:
        continue
    for digit in str(num):
        total_sum += int(digit)

print(total_sum)
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

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

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

n = int(input())

for i in range(1, n + 1):
    if i % 2 == 0:
        continue

    fact = math.factorial(i)
    digit_sum = sum(int(d) for d in str(fact))
    print(f"{i}! = {fact} , sum of digits = {digit_sum}")
```

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

2. 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
N = int(input())

even_sum = 0
odd_sum = 0
for i in range(1, N + 1):
    if i % 2 == 0:
        even_sum += i
    else:
        odd_sum += i
print(f"Sum of even numbers from 1 to {N} is {even_sum}")
print(f"Sum of odd numbers from 1 to {N} is {odd_sum}")
```

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

3. 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
start = int(input())
end = int(input())

total = 0

for num in range(start, end + 1):
   if num % 3 ==0:
      continue
   total += num
print(total)
```

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

4. 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
n = int(input())

steps = 0

while n != 1:
    if steps >= 100:
        print("Exceeded 100 steps. Exiting...")
        break
    if n % 2 == 0:
        n = n // 2
    else:
        n = 3 * n + 1
    steps += 1
else:
    print(f"steps taken to reach 1: {steps}" )
```

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

5. 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
units = int(input())

amount = 0
if units <= 100:
   amount = 0
elif units <=200:
   amount = (units - 100) * 5
else:
   amount = (100 * 5) + (units - 200) * 10
print(f"Rs. {amount}")
```

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

6. 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
def is_leap_year(year):
   return (year %  4 == 0 and year % 100 != 0) or (year % 400 == 0)
start_year = int(input())
end_year = int(input())

for year in range(start_year, end_year + 1):
   if is_leap_year(year):
      print(year)
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Nisha is a mathematics enthusiast, eager to explore the realm of twin prime numbers. The objective is to develop a program that enables the discovery and presentation of twin prime pairs.

The program should take an integer 'n' as input and generate 'n' pairs of twin primes, displaying the pairs with a difference of 2 between them.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the 'n' pairs of twin primes, the pairs with a difference of 2 between them.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

```
Output: 3 5
5 7
11 13
17 19
29 31
```

Answer

```
# You are using Python
def is_prime(num):
    if num <= 1:
        return false
    for j in range(1, int(num**0.5)+ 1):
        if num % j == 0:
            return False
        return True
n = int(input())
count = 0
num = 3


while count < n:
    if is_prime(num) and is_prime(num + 2):
        print(num, num +2)
        count +=1
    num += 1
```

Status : Wrong ***Marks : 0/10***

2. 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

```
# You are using Python
def is_prime(digit):
  return digit in[2, 3, 5, 7]
```

```
X = int(input())

non_prime_sum = 0

for digit_char in str(X):
    digit = int(digit_char)
    if not is_prime(digit):
        non_prime_sum += digit

print(non_prime_sum)
```

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

3. 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

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

first_digit = int(str(n)[0])

last_three_digits = n % 1000

if last_three_digits % first_digit == 0:
   print("Logged In")
else:
   print("Incorrect code")
```

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

4. Problem Statement

Rohith is a data analyst who needs to categorize countries based on their population growth rates. Each country is assigned a unique code. Rohith will receive a code and corresponding data based on the code. If the data falls within specific thresholds, he needs to classify the country's priority level.

Your task is to write a program that reads a country code and its associated data, and then determines if the priority is "High" or "Low."

Thresholds:France: Priority is "High" if the percentage < 50, else "Low".Japan: Priority is "High" if life expectancy > 80, else "Low".Brazil: Priority is "High" if the urban population > 80, else "Low".

Input Format

The first line of input consists of an integer, representing the country code (1 for France, 2 for Japan, 3 for Brazil).

If the country code is 1,

- The second line consists of a floating-point value N, representing the percentage of the English-speaking population.

If the country code is 2,

- The second line consists of a floating-point value A, representing the average life expectancy in years.

If the country code is 3,

- The second line consists of a floating-point value P, representing the percentage of the urban population.

Output Format

The first line of output displays "Priority: High" or "Priority: Low" based on the input data.

If the country code is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
30.0
Output: Priority: High

Answer

```
# You are using Python
try:
    code = int(input())
    if code == 1:
        N = float(input())
        if N < 50:
            print("Priority:\nHigh")
        else:
            print("Priority:\nLow")
    elif code == 2:
        A = float(input())
        if A > 80:
            print("Priority:\nHigh")
        else:
            print("Priority:\nLow")
    elif code == 3:
        P = float(input())
        if P > 80:
            print("Priority:\nHigh")
        else:
            print("Priority:\nLow")
    else:
        print("Invalid")
except:
    print("Invaild")
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_MCQ

Attempt : 1
Total Mark : 25
Marks Obtained : 22

Section 1 : MCQ

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

Answer

Adds a new element to the end of the list

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

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

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

Answer

ValueError: list.remove(x): x not in list

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

3. 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***

4. 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***

5. What is the output of the following code?

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

Answer

6

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

6. 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, 5]

Status : Wrong *Marks : 0/1*

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

Answer

25

Status : Correct *Marks : 1/1*

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

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

Answer

```
**Python**
```

Status : Correct *Marks : 1/1*

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

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

Answer

HelloWorld

Status : Correct *Marks : 1/1*

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

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

Answer

3

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

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

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

Answer

oje

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

12. 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***

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

Answer

list()

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

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

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

Answer

" Python "

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

15. 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***

16. 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***

17. 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***

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

Answer

extend()

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

19. 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***

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

Answer

"%s %s" % (string1, string2)

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

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

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

Answer

Compile Time Error

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

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

Answer

[2, 33, 222, 14]

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

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

Answer

all of the mentioned options

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

24. 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**

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

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

Answer

onhtyP

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 3_COD

Attempt : 1
Total Mark : 50
Marks Obtained : 40

Section 1 : Coding

1. 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

```
# You are using Python
a1=str(input())
a2=a1[1:4]
print("Area Code:",a2)
```

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

2. 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

```
# You are using Python
a1=str(input())
a2=str(input())
print(a1+a2[::-1])
```

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

```
# You are using Python
n=int(input())
l1=[]
for i in range(n):
   a=int(input())
   l1.append(a)
print("List after appending elements:",l1)
m=int(input())
a1=l1.pop(m)
print("List after popping last element:",l1)
print("Popped element:",a1)
```

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

4. 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

```
s1=str(input())
start=int(input())
end=int(input())
if (start<end) and (len(s1)>end):
   print(s1[start:end+1])
else:
   print("Invalid start and end positions")
```

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

5. 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

-

Status : Skipped *Marks : 0/10*

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 14

Section 1 : MCQ

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

```
def maximum(x, y):
    if x > y:
        return x
    elif x == y:
        return 'The numbers are equal'
    else:
        return y

print(maximum(2, 3))
```

Answer

3

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

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

```
def display(b, n):
    while n > 0:
        print(b,end="")
        n=n-1
display('z',3)
```

Answer

zzz

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

3. 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***

4. 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***

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

```
def func(a, b=5, c=10):
    print('a is', a, 'and b is', b, 'and c is', c)

func(3, 7)
func(25, c = 24)
func(c = 50, a = 100)
```

Answer

a is 3 and b is 7 and c is 10a is 25 and b is 5 and c is 24a is 100 and b is 5 and c is 50

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

6. 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***

7. What is the output of the code shown?

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

Answer

worldhellohello

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

8. 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***

9. 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***

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

```
def is_even(number):
   if number % 2 == 0:
      return True

result = is_even(6)
print(result)
```

Answer

True

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

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

```
number = 7
result = abs(number) + pow(number, 2)
```

```
print(result)
```

Answer

56

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

12. What is the output of the code shown below?

```
def f1(x):
   x += 1
   print(x)

global_variable = 15
f1(global_variable)
print("hello")
```

Answer

16hello

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

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

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

Answer

5

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

14. 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***

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

Answer

lambda

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_COD_Updated

Attempt : 1
Total Mark : 50
Marks Obtained : 40

Section 1 : Coding

1. 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

```
s1=input()
print(len(s1))
```

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

2. 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

```
# You are using Python
def is_armstrong_number(number):

   num_str = str(number)
   num_digits = len(num_str)

   armstrong_sum = sum(int(digit) ** num_digits for digit in num_str)

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


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

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

3. 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 base¹ 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

```
# You are using Python
def exponential_calculator(base, exponent):
    # Calculate the final result
    result = pow(base, exponent)

    powers_list = [pow(base, i) for i in range(1, exponent + 1)]

    powers_sum = sum(powers_list)

    print(result)
    print(powers_list)
    print(powers_sum)

base = int(input())
exponent = int(input())

exponential_calculator(base, exponent)
```

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

4. 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):

```
    # Initialize counters for each category
    uppercase_count = 0
    lowercase_count = 0
    digit_count = 0
    special_count = 0

    # Iterate over each character in the input string
    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

    # Print the results in the desired format

# Sample test cases
analyze_string("Hello123")
analyze_string("12345")

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 : Wrong ***Marks : 0/10***

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

```
# Input: Main string and the prefix to check
main_string = input().strip()
prefix = input().strip()

# Lambda function to check if main_string starts with prefix
check_start = lambda s, p: s.startswith(p)

# Output the result
print(check_start(main_string, prefix))
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_PAH_Updated

Attempt : 1
Total Mark : 60
Marks Obtained : 0

Section 1 : Coding

1. 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

-

Status : **-** ***Marks : 0/10***

2. 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

-

Status : - ***Marks : 0/10***

3. 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

-

Status : **-** ***Marks : 0/10***

4. 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

-

Status : - ***Marks : 0/10***

5. 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

-

Status : Skipped ***Marks : 0/10***

6. 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

-

Status : Skipped ***Marks : 0/10***

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 0

Section 1 : Coding

1. 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

-

Status : Skipped ***Marks : 0/10***

2. Problem Statement

Amrita is developing a password strength checker for her website. She wants the checker to consider the length and the diversity of characters used in the password. A strong password should be long and include a mix of character types: uppercase, lowercase, digits, and special symbols.

She also wants the feedback to be user-friendly, so she wants to include the actual password in the output. Help Amrita finish this password checker using Python's built-in string methods.

Character Types Considered:

Lowercase letters (a-z)Uppercase letters (A-Z)Digits (0-9)Special characters (from string.punctuation, e.g. @, !, #, $)

Input Format

The input consists of a single string representing the user's password.

Output Format

The program prints the strength of the password in this format:

If the password length < 6 characters or fewer than 2 of the 4 character types, the output prints "<password> is Weak"

If password length ≥ 6 and at least 2 different character types, the output prints "<password> is Moderate"

If Password length ≥ 10 and all 4 character types present, the output prints "<password> is Strong"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: password123
Output: password123 is Moderate

Answer

-

Status : - *Marks : 0/10*

3. 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

-

Status : - ***Marks : 0/10***

4. 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

```
# You are using Python
# Read input
n_and_numbers = input().split()
```

```
n = int(n_and_numbers[0])
numbers = list(map(int, n_and_numbers[1:n+1]))


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 : Wrong ***Marks : 0/10***

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_MCQ

Attempt : 1
Total Mark : 20
Marks Obtained : 8

Section 1 : MCQ

1. 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

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

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

2. 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***

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

Answer

True

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

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

Answer

print(t[3])

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

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

Answer

Values of a dictionary must be unique

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

6. 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 : Correct ***Marks : 1/1***

7. What will be the output?

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

Answer

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

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

8. 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***

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

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

Answer

Too many arguments for sum() method

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

10. What is the output of the following code?

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

Answer

False

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

11. 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***

12. Which of the following is a Python tuple?

Answer

{1, 2, 3}

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

13. 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***

14. 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

{‘a’: 1, ‘b’: 2, ‘c’: 3}

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

15. Predict the output of the following Python program

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

Answer

{1, 2}(1, 2)

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

16. 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***

17. 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***

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

Answer

set()

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

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

Answer

Keys must be immutable

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

20. What is the output of the following code?

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

Answer

{1: ‘A’, 2: ‘D’, 3: ‘C’}

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 5_COD

Attempt : 2
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. 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

```
# Read number of tuples
N = int(input())

inventory = []
for _ in range(N):
    item = eval(input())
    inventory.append(item)

threshold = int(input())

filtered_quantities = []
for _, quantities in inventory:
    filtered = filter(lambda x: x > threshold, quantities)
    filtered_quantities.extend(filtered)
```

```
print(*filtered_quantities)
```

Status : Correct **_Marks : 10/10_**

2. 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

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

# Read the two lists as comma-separated integers
list1 = list(map(int, input().split(',')))
list2 = list(map(int, input().split(',')))

# Compute the element-wise sum
sum_tuple = tuple(list1[i] + list2[i] for i in range(n))

# Print the resulting tuple
print(sum_tuple)
```

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

3. Problem Statement

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

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

Input Format

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

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

Output Format

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
101 100 150 200
102 50 75 100
Output: {101: [450, 200], 102: [225, 100]}

Answer

```
n = int(input())

customer_data = {}

for _ in range(n):
    data = list(map(int, input().split()))
    customer_id = data[0]
    expenditures = data[1:]
    total = sum(expenditures)
    maximum = max(expenditures)
    customer_data[customer_id] = [total, maximum]

print(customer_data)
```

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

4. Problem Statement

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

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

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

Input Format

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

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

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

Output Format

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3
2 3 4
3 4 5

Output: {2, 3}

{2}

Answer

```
# Read input
registered = set(map(int, input().split()))
attended = set(map(int, input().split()))
dropped_out = set(map(int, input().split()))

# Students who registered and attended
registered_and_attended = registered & attended

# Students who attended and did not drop out
final_participants = registered_and_attended - dropped_out

# Print results
print(registered_and_attended, final_participants)
```

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

5. 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

```
# Read number of product IDs
n = int(input())

# Initialize frequency dictionary
frequency = {}

# Read each product ID and update frequency
for _ in range(n):
    pid = int(input())
    if pid in frequency:
        frequency[pid] += 1
    else:
        frequency[pid] = 1

# Calculate total unique IDs
total_unique = len(frequency)

# Calculate total occurrences
total_occurrences = sum(frequency.values())

# Calculate average frequency
average_frequency = total_occurrences / total_unique

# Print outputs
print(frequency)
print(f"Total Unique IDs: {total_unique}")
print(f"Average Frequency: {average_frequency:.2f}")
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_MCQ

Attempt : 1
Total Mark : 20
Marks Obtained : 4

Section 1 : MCQ

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

Answer

depends on the operating system

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

2. 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, 2)2) f.readlines()

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

3. How do you rename a file?

Answer

os.rename(fp, new_name)

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

4. 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-iv, d-ii

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

5. 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 : Wrong ***Marks : 0/1***

6. Which of the following is true about

fp.seek(10,1)

Answer

Move file pointer ten characters behind from the current position

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

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

Answer

raise Exception()

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

8. 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

This will always execute

Status : Wrong *Marks : 0/1*

9. 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,w,pickle,r)

Status : Wrong *Marks : 0/1*

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

Answer

To define a custom exception

Status : Wrong *Marks : 0/1*

11. What is the output of the following code?

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

```
finally:
   print("Executed")
```

Answer

Executed

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

12. 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

Type Error occurred

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

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

Answer

raise CustomException()

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

14. 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***

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

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

Answer

null

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

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

Answer

The finally block is executed only if no exception occurs

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

17. 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***

18. 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***

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

Answer

file position is set to the end of file

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

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

Answer

except

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

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

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

```
# You are using Python
file_name = "text_file.txt"

input_string = input()

with open(file_name, "w") as file:
   file.write(input_string)

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

print(f"Original String: {original_string}")
print(f"Upper-Case String: {original_string.upper()}")
print(f"Lower-Case String: {original_string.lower()}")
```

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

2. 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

# You are using Python

```
def get_integer_input(prompt):
    try:
        value = int(input(prompt))
        return value
    except ValueError:
        print("Error: You must enter a numeric value.")
        exit()

n = get_integer_input("")

if n <= 0 or n > 20:
    print("Error: The length of the list must be a non-negative integer.")
    exit()

numbers = []
for _ in range(n):
    num = get_integer_input("")
    numbers.append(num)

average = sum(numbers) / n
print(f"The average is: {average:.2f}")
```

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

3. 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

```
# You are using Python
file_name = "sentence_file.txt"

input_sentence = input()

with open(file_name, "w") as file:
   file.write(input_sentence)

with open(file_name, "r") as file:
   sentence = file.read()

word_count = len(sentence.split())

print(word_count)
```

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

4. 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

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

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

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

5. 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

```
# You are using Python
price = float(input())
quantity = int(input())

try:
   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 e:
   print(e)
except RuntimeError as e:
   print(e)
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_PAH

Attempt : 1
Total Mark : 30
Marks Obtained : 28.5

Section 1 : Coding

1. 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

```
MAX_CAPACITY = 30
student_database = {}

n = int(input())

for _ in range(n):
    student_id, student_name = input().split(maxsplit=1)
    student_id = int(student_id)
    if len(student_database) >= MAX_CAPACITY:
        print("Exception caught. Error: Student database is full.")
        break
    elif student_id in student_database:
        print("Exception caught. Error: Student ID already exists.")
    else:
        student_database[student_id] = student_name
        print(f"Student with ID {student_id} added to the database.")
```

Status : Partially correct ***Marks : 8.5/10***

2. 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

```
file_name = "user_input.txt"

user_input = input()
with open(file_name, "w") as file:
    file.write(user_input)

try:
    with open(file_name, "r") as file:
        data = file.read().strip().split()

    numbers = [float(item) for item in data]

    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***

3. 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

```
file_name = input()
text_content = input()
char_to_count = input()

with open(file_name, "w") as file:
   file.write(text_content)

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

char_count = content.upper().count(char_to_count.upper())

if char_count > 0:
   print(f"The character '{char_to_count}' appears {char_count} times in the file.")
else:
   print("Character not found in the file.")
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 6_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

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

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

Input Format

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

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

Output Format

The output should display any one of the following messages:

If both numbers are valid, print "Valid".

If an exception is raised, print "Invalid with exception message: ", followed by the specific exception message.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 19ABC1001
9949596920

Output: Valid

Answer

```
import re

class IllegalArgumentException(Exception):
    pass

class NumberFormatException(Exception):
    pass

class NoSuchElementException(Exception):
    pass

def validate_register_and_mobile(register_number, mobile_number):
    try:

        if len(register_number) != 9:
            raise IllegalArgumentException("Register Number should have exactly 9 
characters.")

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

        if len(mobile_number) != 10:
            raise IllegalArgumentException("Mobile Number should have exactly 10 
characters.")

        if not mobile_number.isdigit():
            raise NumberFormatException("Mobile Number should only contain 
digits.")

        print("Valid")

    except IllegalArgumentException as e:
        print(f"Invalid with exception message: {e}")
    except NumberFormatException as e:
        print(f"Invalid with exception message: {e}")
    except NoSuchElementException as e:
        print(f"Invalid with exception message: {e}")

register_number = input()
mobile_number = input()

validate_register_and_mobile(register_number, mobile_number)
```

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

2. Problem Statement

Alice is developing a program called "Name Sorter" that helps users organize and sort names alphabetically.

The program takes names as input from the user, saves them in a file, and then displays the names in sorted order.

File Name: sorted_names.txt.

Input Format

The input consists of multiple lines, each containing a name represented as a string.

To end the input and proceed with sorting, the user can enter 'q'.

Output Format

The output displays the names in alphabetical order, each name on a new line.

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: Alice Smith
John Doe
Emma Johnson
q
Output: Alice Smith
Emma Johnson
John Doe
```

Answer

```
# You are using Python
def name_sorter():
    names = []

    while True:
        name = input(" ").strip()
        if name.lower() == 'q':
            break
        if len(name) >= 3 and len(name) <= 30:
            names.append(name)

    names.sort()

    with open("sorted_names.txt", "w") as file:
        for name in names:
            file.write(name + "\n")

    for name in names:
        print(name)

name_sorter()
```

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

```
# You are using Python
def calculate_gpa(grades):

    return sum(grades) / len(grades)

def main():
    magical_grades = []
    while True:

        student_name = input(" ")

        if student_name.lower() == 'done':
            break

        subjects_and_grades = []
        for _ in range(2):
            subject = input(f" ")
            grade = float(input(f" "))

            if 0 <= grade <= 100:
                subjects_and_grades.append(grade)
            else:
                print("Invalid grade. Please enter a grade between 0 and 100.")
                return


        gpa = calculate_gpa(subjects_and_grades)
        magical_grades.append((student_name, subjects_and_grades, gpa))


        with open("magical_grades.txt", "a") as file:
            file.write(f"Student: {student_name}\n")
            file.write(f"Subjects and Grades: {subjects_and_grades}\n")
            file.write(f"GPA: {gpa:.2f}\n\n")

        print(f" {gpa:.2f}\n")


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

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

4. 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

```
# You are using Python
from collections import Counter

def analyze_char_frequency(input_string):

 char_count = Counter(input_string)
```

```
    with open("char_frequency.txt", "w") as file:

        file.write("\n")

        for char, count in char_count.items():
            file.write(f"{char}: {count}\n")

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

input_string = input(" ")

analyze_char_frequency(input_string)
```

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 7_MCQ

Attempt : 1
Total Mark : 20
Marks Obtained : 19

Section 1 : MCQ

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

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

Answer

[1 2 3 1 2 3]

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

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

Answer

numpy.std()

Status : Correct *Marks : 1/1*

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

Answer

Creates an array with values from 1 to 9

Status : Correct *Marks : 1/1*

4. The important data structure of pandas is/are ____.

Answer

Both Series and Data Frame

Status : Correct *Marks : 1/1*

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

Answer

arr[0]

Status : Correct *Marks : 1/1*

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

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

Answer

```
[3 4 5]
```

Status : Correct *Marks : 1/1*

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

Answer

argmax() and argmin()

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

8. What is the output of the following code?

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

Answer

[2, 3, 4]

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

9. What is the purpose of the following NumPy code snippet?

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

Answer

Displays a 3x4 matrix filled with zeros

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

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

Answer

1

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

11. Which NumPy function is used to create an identity matrix?

Answer

numpy.eye()

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

12. What does NumPy stand for?

Answer

Numerical Python

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

13. Which function is used to create a Pandas DataFrame?

Answer

pd.DataFrame()

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

14. What is the result of the following NumPy operation?

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

Answer

[6 7 8]

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

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

```
import pandas as pd

data = {'Name': ['Tom', 'Jack', 'nick', 'juli'],
 'marks': [99, 98, 95, 90]}

df = pd.DataFrame(data, index=['rank1',
           'rank2',
           'rank3',
           'rank4'])
print(df)
```

Answer

rank2

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

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

Answer

import numpy as np

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

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

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

Answer

[3 4]

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

18. What is the primary purpose of Pandas DataFrame?

Answer

To store data in tabular form for analysis and manipulation

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

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

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

Answer

Value Error

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

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

Answer

Array

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

Rajalakshmi Engineering College

Name: kavin v
Email: 240701242@rajalakshmi.edu.in
Roll no: 240701242
Phone: 8248033180
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 7_COD

Attempt : 1
Total Mark : 50
Marks Obtained : 47.5

Section 1 : Coding

1. Problem Statement

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

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

Implement this data cleaning task using the pandas package.

Input Format

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

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

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

Output Format

The first line of output is the text:

Cleaned Hospital Records:

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

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 5
PatientID,Name,Age,Diagnosis
1,John Doe,45,Flu
2,Jane Smith,,Cold
3,Bob Lee,50,
4,Alice Green,38,Fever
5,Tom Brown,,Infection
Output: Cleaned Hospital Records:
   PatientID         Name        Age  Diagnosis
0          1     John Doe  45.000000        Flu
1          2   Jane Smith  44.333333       Cold
2          4  Alice Green  38.000000      Fever
3          5    Tom Brown  44.333333  Infection
```

Answer

```
import pandas as pd
import sys
import numpy as np

# Read input
n = int(input())
header = input().split(',')

# Read the remaining n lines and store in a list
data = [input().split(',') for _ in range(n)]

# Create DataFrame
df = pd.DataFrame(data, columns=header)

df['Age'] = pd.to_numeric(df['Age'], errors='coerce')

mean_age = df['Age'].mean()

df['Age'] = df['Age'].fillna(mean_age)

df['Diagnosis'].replace('', np.nan, inplace=True)
df.dropna(subset=['Diagnosis'], inplace=True)

cleaned_df = df.reset_index(drop=True)

# Output
print("Cleaned Hospital Records:")
print(cleaned_df)
```

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

2. Problem Statement

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

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

Input Format

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

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

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

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

Output Format

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

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

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
NYC NYC LA LA
Jan Feb Jan Feb
100 200 300 400

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

Total Sales Per City:
      Sales
City
```

```
LA    700.0
NYC   300.0
```

Answer

```
import pandas as pd

# Read input
n = int(input())  # Number of records

cities = input().split()
months = input().split()
sales = list(map(float, input().split()))

# Create MultiIndex from city and month
index = pd.MultiIndex.from_tuples(zip(cities, months), names=["City", "Month"])

# Create DataFrame with sales data
df = pd.DataFrame({"Sales": sales}, index=index)

# Print the MultiIndex DataFrame
print("Monthly Sales Data with MultiIndex:")
print(df)

# Group by City and calculate total sales
total_sales = df.groupby(level='City').sum()

# Print total sales per city
print("\nTotal Sales Per City:")
print(total_sales)
```

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

3. Problem Statement

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

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

Input Format

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

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

Output Format

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 2 4
10 20 30 40
5 15 25 35

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

Answer

```
import numpy as np

# Read number of products and months
products, months = map(int, input().split())

# Read sales data for each product
sales_data = [list(map(int, input().split())) for _ in range(products)]

# Convert to NumPy array
sales_array = np.array(sales_data)
```

```
# Compute cumulative sum along each row (i.e., across months)
cumulative_array = np.cumsum(sales_array, axis=1)

# Output results
print("Cumulative Monthly Sales:")
print(cumulative_array)
```

Status : Correct **_Marks : 10/10_**

4. Problem Statement

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

Help her to implement the task using the numpy package.

Formula:

To filter sales values:

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

Input Format

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

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

Output Format

The output prints: filtered_sales

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
50.0 100.0 105.0 150.0 99.0
Output: [105. 150.]

Answer

```
import numpy as np

# Read input
n = int(input())
sales = np.array(list(map(float, input().split())))

# Apply filter: multiples of 5 and greater than 100
filtered_sales = sales[(sales % 5 == 0) & (sales > 100)]

# Print result
print(filtered_sales)
```

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

5. Problem Statement

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

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

Input Format

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

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

Output Format

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1.0 2.0 3.0
4.0 5.0 6.0
Output: Cross-correlation of the two time series:
[6. 17. 32. 23. 12.]

Answer

```
import numpy as np

# Read inputs
array1 = np.array(list(map(float, input().split())))
array2 = np.array(list(map(float, input().split())))

cross_corr = np.correlate(array1, array2, mode='full')

print("Cross-correlation of the two time series:")
print(cross_corr)
```

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