

Rajalakshmi Engineering College

Name: talapaneni sujitha
Email: 240701551@rajalakshmi.edu.in
Roll no: 2116240701551
Phone: 9445179891
Branch: REC
Department: I CSE FF
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 : 9

Section 1 : MCQ

1. What is the purpose of the pass statement in Python?

Answer

To do nothing and act as a placeholder.

Status : Correct

Marks : 1/1

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

```
i = 5
while True:
    if i%0011 == 0:
        break
    print(i)
    i += 1
```

Answer

5 6 7 8 9 10

Status : Wrong

Marks : 0/1

3. 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

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

```
i = 5
while True:
    if i%0011 == 0:
        break
    print(i, end = " ")
    i += 1
```

Answer

5 6 7 8

Status : Correct

Marks : 1/1

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

```
i = 1
while True:
    if i%007 == 0:
```

```
break
print(i)
i += 1
```

Answer

1 2 3 4 5 6

Status : Correct

Marks : 1/1

6. What is the output of the following?

```
True = False
while True:
    print(True)
    break
```

Answer

False

Status : Wrong

Marks : 0/1

7. 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

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

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

Answer

none of the above mentioned options

Status : Wrong

Marks : 0/1

9. 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

10. 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

11. What is the output of the following code?

```
i = 5
while True:
    if i%1009 == 0:
        break
    print(i)
    i += 1
```

Answer

5 6 7 8 9

Status : Wrong

Marks : 0/1

12. What is the output of the following program?

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

Answer

1 1 1 2 2 2

Status : Correct

Marks : 1/1

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

```
i = 1
while True:
    if i % 2 == 0:
        i += 1
        continue
```

```
if i > 10:  
    break  
print(i, end = " ")  
i += 2
```

Answer

1 3 5 7 9 11

Status : Wrong

Marks : 0/1

14. What will the following code output?

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

Answer

3

Status : Correct

Marks : 1/1

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

Answer

break

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: talapaneni sujitha
Email: 240701551@rajalakshmi.edu.in
Roll no: 2116240701551
Phone: 9445179891
Branch: REC
Department: I CSE FF
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

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

```
n1=int(input())
n2=int(input())
total_sum=0
for i in range(n1,n2+1):
    str_i=str(i)
    if str_i!=str_i[::-1]:
```



```
digit_sum=sum(int(digit)for digit in str_i)
total_sum+=digit_sum
print(total_sum)
```

Status : Correct

Marks : 10/10

2. Problem Statement

You work as an instructor at a math enrichment program, and your goal is to develop a program that showcases the concept of using control statements to manipulate loops. Your task is to create a program that takes an integer 'n' as input and prints the squares of even numbers from 1 to 'n', while skipping odd numbers.

Input Format

The input consists of a single integer, which represents the upper limit of the range.

Output Format

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10

Output: 4

16

36

64

100

Answer

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

pass

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

```
x=int(input())
y=int(input())
for i in range(x,y):
    total_sum=0
    new_r=i//2
```

```
for j in range(1,new_r+1):
    if i%j==0:
        total_sum+=j
    if i==total_sum:
        print(i,end=" ")
```

Status : Correct

Marks : 10/10

4. 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

```
word=str(input())
vowel='aeiouAEIOU'
for i in word:
    if i.isalpha() and i not in vowel:
        print(i,end=" ")
```

else:
pass

Status : Correct

Marks : 10/10

5. Problem Statement

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

Help Emma determine the count of non-Fibonacci numbers within the given range [start, end] using the continue statement.

Input Format

The first line of input consists of an integer, representing the starting number of the range.

The second line consists of an integer, representing the ending number of the range.

Output Format

The output prints a single integer, representing the count of numbers in the range that are not Fibonacci numbers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
10

Output: 5

Answer

```
n1=int(input())
n2=int(input())
a,b=0,1
fib=set()
while a<=n2:
```

```
fib.add(a)
a,b=b,a+b
count=0
for num in range(n1,n2+1):
    if num not in fib:
        count+=1
print(count)
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: talapaneni sujitha
Email: 240701551@rajalakshmi.edu.in
Roll no: 2116240701551
Phone: 9445179891
Branch: REC
Department: I CSE FF
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 : 50

Section 1 : Coding

1. 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

```
def manage_list():
```

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

```
    elem=[]
```

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

```
        ele=int(input())
```

```
        elem.append(ele)
```

```
    M=int(input())
```

```
    print(f"List after appending elements: {elem}")
```

```
    popele = elem.pop(M)
```

```
    print(f"List after popping last element: {elem}")
```

```
    print(f"Popped element: {popele}")
```

manage_list()

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
str1=input()
str2=input()
print(str1+str2[::-1])
```

Status : Correct

Marks : 10/10

3. 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

```
# You are using Python
```

```
s=input()
```

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

```
m=int(input())
```

```
if(m<=n or n>len(s)):
```

```
    print("Invalid start and end positions")
```

```
else:
```

```
s1=s[n:m+1]  
print(s1)
```

Status : Correct

Marks : 10/10

4. 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

```
# You are using Python
str1=input()
arr=list(map(int,str1.strip('[]').split(',')))
negatives=[x for x in arr if x<0]
non_negatives=[x for x in arr if x>=0]
res=negatives+non_negatives
print("List=",res)
```

Status: Correct

Marks : 10/10

5. 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

```
s=input()
```

```
print(f"Area code: {s[1:4]}")
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: talapaneni sujitha
Email: 240701551@rajalakshmi.edu.in
Roll no: 2116240701551
Phone: 9445179891
Branch: REC
Department: I CSE FF
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

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

```
def generator(n):
    count = 0
    while n!=1:
        if n%2==0 and count<=100:
            n//=2
            count+=1
        elif n%2==1 and count<=100:
            n=(n*3)+1
            count+=1
        else:
            print("Exceeded 100 steps.exiting...")
            return
    print(f"steps taken to reach 1:{count}")
    return
num=int(input())
generator(num)
```

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

```
def generator(n):
    oddsum=0
    evensum=0
    for i in range(1,n+1):
        if i%2==0:
            evensum+=i
        else:
            oddsum+=i
    print(f"Sum of even numbers from 1 to {n} is {evensum}")
    print(f"Sum of odd numbers from 1 to {n} is {oddsum}")
    return
num=int(input())
generator(num)
```

Status : Correct

Marks : 10/10

3. Problem Statement

As a software engineer, your goal is to develop a program that facilitates the identification of leap years in a specified range. Your task is to create a program that takes two integer inputs, representing the start and end years of the range and then prints all the leap years within that range.

Input Format

The first line of the input consists of an integer, which represents the start year.

The second line consists of an integer, which represents the end year.

Output Format

The output displays the leap years within the given range, separated by lines.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2000

2053

Output: 2000

2004

2008

2012

2016

2020

2024

2028

2032

2036

2040

2044

2048

2052

Answer

```
def generator(a,b):  
    for i in range(a,b+1):  
        if(i%4==0 and i%100!=0) or (i%400==0):  
            print(i)  
    return  
y1=int(input())  
y2=int(input())  
generator(y1,y2)
```

Status : Correct**Marks : 10/10****4. 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

```
def generator(n1,n2):
    tot=0
    for i in range(n1,n2+1):
        if i%3!=0:
            tot+=i
    print(tot)
    return
num1=int(input())
num2=int(input())
generator(num1,num2)
```

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

```
def generator(n):  
    if n<=100:  
        print("Rs.0")  
        return  
    if n>100 and n<=200:  
        pay=(n-100)*5  
        print(f"Rs.{pay}")  
        return  
    if n>200:  
        pay=(n-200)*10+(100*5)  
        print(f"Rs.{pay}")  
        return  
num=int(input())  
generator(num)
```

Status : Correct

Marks : 10/10

6. 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

```
def fact(f_n):
    if f_n == 1 or f_n == 0:
        return 1
    return f_n*fact(f_n-1)

def sumofdig(s_n):
    s_n = str(s_n)
    digitsum = sum(int(i) for i in s_n)
    return digitsum

def generator(n):
    for i in range(1 ,n+1):
        if i % 2 == 1:
            factorial = fact(i)
            digit_sum = sumofdig(factorial)
            print(f"{i}! = {factorial}, sum of digits = {digit_sum}")
    return

num = int(input())
generator(num)
```

Status : Correct

Marks : 10/10