Rajalakshmi Engineering College

Name: Swetha R
Email: 240701550@rajalakshmi.edu.in
Roll no: 240701550
Phone: 9444370187
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 : 10

Section 1 : MCQ

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

```
balloon_inflated = False
while not balloon_inflated:
    if not balloon_inflated:
        balloon_inflated = True
        print("inflate-", end="")
print("done")
```

Answer

inflate-done

Status : Correct   *Marks : 1/1*

2. What is the output of the following?

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

Answer

0 1 2 3 4 Here

Status : Wrong   *Marks : 0/1*

3. 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*

4. 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*

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

Answer

When loop condition becomes false

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, end = " ")
    i += 2
```

Answer

1 3 5 7 9

Status : Correct   *Marks : 1/1*

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

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

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

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

Answer

012

Status : Correct

Marks : 1/1

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

Answer

break

Status : Correct

Marks : 1/1

10. What is the output of the following code?

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

Answer

0 1 2 3 4

Status : Wrong   *Marks : 0/1*

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

Answer

To do nothing and act as a placeholder.

Status : Correct   *Marks : 1/1*

12. 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

02

Status : Correct   *Marks : 1/1*

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

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

Answer

5 6 7 8 9 10

Status : Wrong   *Marks : 0/1*

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

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

Answer

The code runs successfully but does not print anything

Status : Correct   *Marks : 1/1*

15.   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*

Rajalakshmi Engineering College

Name: Swetha R
Email: 240701550@rajalakshmi.edu.in
Roll no: 240701550
Phone: 9444370187
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

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

```python
# You are using Python
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

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

```
    else:
        pass
```

3. 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(2,n+1,2):
    print(i*i)
```

Status : Correct   *Marks : 10/10*

4. 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

```python
# You are using Python
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*

5. 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
x=int(input())
y=int(input())
```

```python
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*

Rajalakshmi Engineering College

Name: Swetha R
Email: 240701550@rajalakshmi.edu.in
Roll no: 240701550
Phone: 9444370187
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

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

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

```python
# You are using Python
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*

2. Problem Statement

Sophia, a primary school teacher, wants to calculate the sum of numbers within a given range, excluding those that are multiples of 3.

Write a program to help Sophia compute the sum of all numbers between

start and end (inclusive) that are not divisible by 3 using the continue statement.

Input Format

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

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

Output Format

The output prints a single integer, representing the sum of numbers in the range that are not multiples of 3.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
10
Output: 37

Answer

```python
# You are using Python
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*

3. 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

```python
# You are using Python
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*

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

```python
# You are using Python
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*

5. 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

```python
# You are using Python
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*

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

```python
# You are using Python
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)
```

```python
        print(f"{i}!={factorial},sum of digits = {digit_sum}")
    return

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

Status : Correct   *Marks : 10/10*

Rajalakshmi Engineering College

Name: Swetha R
Email: 240701550@rajalakshmi.edu.in
Roll no: 240701550
Phone: 9444370187
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_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1.   Problem Statement

Taylor is tasked with a mathematical challenge that requires finding the smallest positive number divisible by all integers from 1 to n.

Help Taylor to determine the smallest positive number that is divisible by all integers from 1 to n. Make sure to employ the break statement to ensure efficiency in the program.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the smallest positive number that is divisible by all integers from 1 to n.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10
Output: 2520

Answer

```python
# You are using Python
import math
def generator(n):
    lcm=1
    for i in range(1,n+1):
        lcm= lcm*i//math.gcd(lcm,i)
    print(lcm)

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

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

2.   Problem Statement

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

Conditions:

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

Write a program to help Gabriel analyze the characteristics of animals

based on the given inputs.

Input Format

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

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

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

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

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

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

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

Output Format

The output prints one of the following:

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

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

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

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

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

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

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: d
2.5
Output: Distance: 857.50 m

Answer

```python
# You are using Python
ch=str(input())
if(ch=='D' or ch=='d'):
    t=float(input())
    dis= float(343*t)
    print("Distance: {:.2f}m".format(dis))
elif(ch=='B' or ch=='b'):
    ft=float(input())
    w= float(5*ft)
    print("Weight: {:.2f}lb".format(w))
elif(ch=='F' or ch=='f'):
    p=float(input())
    d=float(input())
    alt=p*d
    print("Altitude: {:.2f}m".format(alt))
else:
    print("Invalid")
```

Status : Correct   *Marks : 10/10*

3. Problem Statement

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

prime digits in a given integer using loops.

Help Alex to complete his task.

Example:

Input:

845

output:

12

Explanation:

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

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

Output: 12

Input Format

The input consists of a single integer X.

Output Format

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 845
Output: 12

Answer

```python
# You are using Python
def isprime(num):
    if num<2:
        return False
    for i in range(2,int(num**0.5)+1):
```

```
    if num%i==0:
        return False
    return True

def generator(n):
    tot=0
    while n!=0:
        if not isprime(n%10):
            tot=tot+(n%10)
        n=n//10
    print(tot)

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

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

4. Problem Statement

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

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

Input Format

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

Output Format

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

Output: 2 3 5 13 89

Answer

```python
# You are using Python
def is_prime(num):
    if num<2:
        return False
    for i in range(2,int(num**0.5)+1):
        if num %i ==0:
            return False
    return True
def prime_fibonacci(n):
    fib1,fib2=0,1
    prime_fib_numbers=[]

    while len(prime_fib_numbers)<n:
        fib_next=fib1+fib2
        if is_prime(fib_next):
            prime_fib_numbers.append(fib_next)
        fib1,fib2=fib2,fib_next
    print(" ".join(map(str,prime_fib_numbers)))

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

Status : Correct   *Marks : 10/10*