|  |  |
| --- | --- |
| **Ex. No. 3**  **Date:19.05.2021** | **CONDITIONAL CONSTRUCTS – LEVEL 1** |

**AIM:**

To write Python programs using branching and looping statements.

**PROGRAMMING BASE:**

**I. Branching Statements**

|  |  |
| --- | --- |
| **Type** | **Description** |
| Simple if | Display the True set of statements if the condition is True. |
| if…else | If the condition is True, True block will be executed; else False block will be executed. |
| elif ladder | If the condition is True, True block will be executed; else the corresponding conditions will be checked. If every condition is False, then False block will be executed. |
| Nested if | If statements are nested. |

1. Simple if

Syntax:

if condition:

# True Block

# Statement(s)

1. if…else

Syntax:

if condition:

# True Block

else:

# False Block

# Statement(s)

1. elif ladder

Syntax:

if condition:

# True Block

elif condition:

# True Block

elif condition:

# True Block

else condition:

# False Block

# Statement(s)

1. Nested if

Syntax:

if condition:

if condition:

if condition:

# True Block

else:

# False Block

else:

# False Block

else:

# False Block

# Statement(s)

**II. Looping Statements**

|  |  |
| --- | --- |
| **Loop Type** | **Description** |
| while loop | Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body. |
| for loop | Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. |

1. While loop

Syntax:

while expression:

# Statement(s)

1. For loop

Syntax:

foriterating\_varin sequence:

# Statements(s)

**PROGRAMS:**

**a) To find GCD of two given numbers**

**Description:**

GCD (Greatest Common Divisor) or HCF (Highest Common Factor) is the largest positive integer that divides each of the integers without leaving any remainder.

Input Format:

Input consists of two integers for which GCD is to be calculated

Output Format:

Output returns one argument, which is the GCD.

Sample Input

24

36

Sample Output

12

**Program:**

**‘’’**Name: R.sridevi

Roll.No: 20UIT021

Program name: To find GCD of two given numbers.’’’

#variable declaration

number\_1=float(input())

number\_2=float(input())

i=1

while(i<=number\_1 and i<=number\_2):

if(number\_1 % i==0 and number\_2 % i==0):

gcd=i

i=i+1

#output declaration

print(gcd)

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 24  36 | 12 |
| 2 | 18  24 | 6 |
| 3 | 18  25 | 1 |
| 4 | 122  44 | 2 |
| 5 | 345  165 | 15 |
| **Total Test Cases** | | **5** |
| **Number of Test Cases Passed** | | **5** |

**b) To find LCM of two given numbers**

**Description:**

LCM (Least Common Multiple) of two numbers is the smallest number which can be divided by both numbers.

Constraints:

The numbers should be only positive numbers

Input Format:

Input consists of 2 integers

Output Format:

Print the LCM of a given number

Sample Input

24

36

Sample Output

72

Sample Input

-98

76

Sample Output

The number is negative

**Program:**

‘’’Name: R.sridevi

Roll.No: 20UIT021

Program name: To find LCM of two given numbers.’’’

#variable declaration

num1=int(input())

num2=int(input())

if(num1>num2):

maximum=num1

else:

maximum=num2

if num1>0 and num2>0:

while(True):

if(maximum % num1==0 and maximum % num2==0):

print(maximum)

break;

maximum=maximum+1

else:

#output statement

print('The number is negative')

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 24  36 | 72 |
| 2 | -98  76 | The number is negative |
| 3 | 24  6 | 24 |
| 4 | 24  15 | 120 |
| **Total Test Cases** | | **4** |
| **Number of Test Cases Passed** | |  |

**RESULT:**

Thus, the Python programs are executed successfully.