Programming in Python (CSE 3142)

MINOR ASSIGNMENT-4: DEBUGGING

1. Consider the following Python code intended to compute the sum of n natural numbers. During testing, it was found that sum printed by program always excludes the last number. Debug the following script using the debugger.
   Program to compute the sum of n natural numbers

```
def summation(n):
    '''
    Objective: To find sum of first n positive integers
    Input Parameter: n - numeric value
    Return Value: total - numeric value
    '''
    total = 0
    for count in range(1, n):
        total += count
    return total

def main():
    '''
    Objective: To find sum of first n positive integers based on user
    input
    Input Parameter: None
    Return Value: None
    '''
    n = int(input('Enter number of terms: '))
    total = summation(n)
    print('Sum of first', n, 'positive integers: ', total )

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

2. Consider the following Python code intended to print inverse right triangle for given numbers of rows nRows. For example, for nRows = 5, the following inverted triangle should be printed:

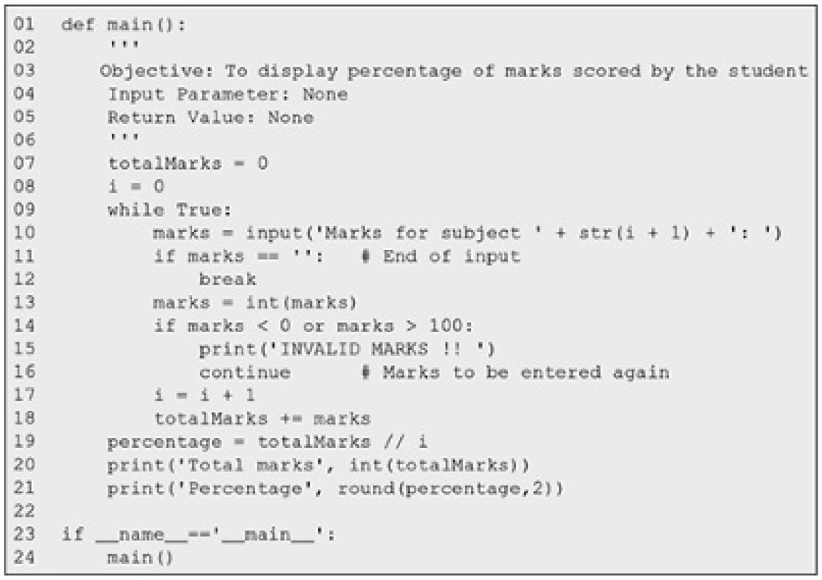
```
*****
****
***
**
*
```

   During testing, it was found that the program does not produce even the single line of output. Debug the following script using the debugger.
   Program to print inverse right triangle

```
def invertedRightTriangle(nRows):
    '''
    Objective: To print right triangle
    Input Parameter: nRows - integer value
    Return Value: None
    '''
    for i in range(nRows, 0):
        print('*' * i)

def main():
    '''
    Objective: To print right triangle
    Input Parameter: None
    Return Value: None
    '''
    nRows = int(input('Enter no. of rows: '))
    invertedRightTriangle(nRows)

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

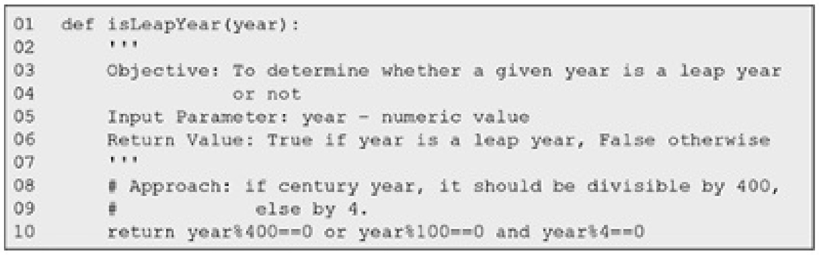
3. Consider the Python script given below intended to compute the percentage. During testing, it was found that percentage computed was not accurate rather rounded to lower bound integer value. Debug the following script using the debugger.

Program to print inverse right triangle



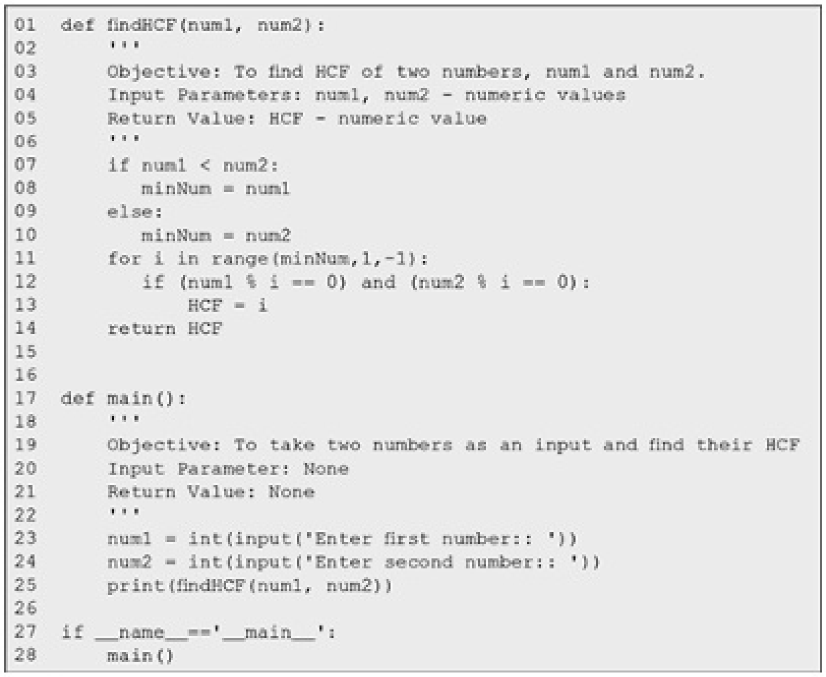
```
def main():
    '''
    Objective: To display percentage of marks scored by the student
    Input Parameter: None
    Return Value: None
    '''
    totalMarks = 0
    i = 0
    while True:
        marks = input('Marks for subject ' + str(i + 1) + ': ')
        if marks == '':    # End of input
            break
        marks = int(marks)
        if marks < 0 or marks > 100:
            print('INVALID MARKS !! ')
            continue       # Marks to be entered again
        i = i + 1
        totalMarks += marks
    percentage = totalMarks // i
    print('Total marks', int(totalMarks))
    print('Percentage', round(percentage,2))

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

4. Consider the Python given below intended to determine whether the given year is a leap year. During testing, it was found that an year such as 1800 or 2100, despite being non-leap year, was also displayed as a leap-year. Debug the following script using the debugger.
   Program to print inverse right triangle



```
def isLeapYear(year):
    '''
    Objective: To determine whether a given year is a leap year
               or not
    Input Parameter: year - numeric value
    Return Value: True if year is a leap year, False otherwise
    '''
    # Approach: if century year, it should be divisible by 400,
    #             else by 4.
    return year%400==0 or year%100==0 and year%4==0
```

5. Consider the Python script given below intended to find HCF. During testing, it was found that program yields an error for numbers having no common factor other than 1. Debug the following script using the debugger.
   Program to print inverse right triangle



```
def findHCF(num1, num2):
    '''
    Objective: To find HCF of two numbers, num1 and num2.
    Input Parameters: num1, num2 - numeric values
    Return Value: HCF - numeric value
    '''
    if num1 < num2:
       minNum = num1
    else:
       minNum = num2
    for i in range(minNum,1,-1):
       if (num1 % i == 0) and (num2 % i == 0):
           HCF = i
    return HCF


def main():
    '''
    Objective: To take two numbers as an input and find their HCF
    Input Parameter: None
    Return Value: None
    '''
    num1 = int(input('Enter first number:: '))
    num2 = int(input('Enter second number:: '))
    print(findHCF(num1, num2))

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