Aim: Write python programs to implement different expressions, variables, Quotes, Basic Math operations, decision making and looping statements etc.

Write python programs

a. To compute prime factors of an integer.
b. To Print the Pascal's triangle for n number of rows given by the user.
c. To Find all Numbers in a Range which are Perfect Squares and Sum of all Digits in the Number is Less than 10

Objective of the Experiment:

1. Understanding expressions, variables, Math operations, control structures etc.

Algorithms:

a. Algorithm to compute prime factors:

1. Take the value of the integer and store in a variable.
2. Using a while loop, first obtain the factors of the number.
3. Using another while loop within the previous one, compute if the factors are prime or not.
4. Exit.

b. Algorithm to print the Pascal's triangle for n number of rows given by the user:

1. Take in the number of rows the triangle should have and store it in a separate variable n.
2. Using a for loop which ranges from 0 to n-1, append the sub-lists into the list.
3. Then append 1 into the sub-lists.
4. Then use a for loop to determine the value of the number inside the triangle.
5. Print the Pascal's triangle according to the format.
6. Exit.

c. Algorithm to find all numbers in a range which are perfect squares and sum of all digits in the number is less than 10.

1. User must enter the upper and lower range for the numbers.
2. A list must be created using list comprehension where the element is a perfect square within the range and the sum of the digits of the number is less than 10.
3. This list must then be printed.
4. Exit

Source code for the implementation:

(Write only important functions)

Post Lab Assignment:

1. One of the following 10 statements generates an error. Which one? (Your answer should be a number between 1 and 10.)

```
x = [1,"abcd",2,"efgh",[3,4]]  # Statement 1
y = x[0:50]                    # Statement 2
z = y                          # Statement 3
w = x                          # Statement 4
x[1] = x[1][0:3] + 'd'         # Statement 5
y[2] = 4                       # Statement 6
z[0] = 0                       # Statement 7
x[1][1:2] = 'yzw'              # Statement 8
w[4][0] = 1000                 # Statement 9
a = (x[4][1] == 4)             # Statement 10
```

2. What would happen if we call gcd(m,n) with m positive and n negative in the following definition?

```
def gcd(m,n):
  if m < n:
    (m,n) = (n,m)
  if (m % n) == 0:
    return(n)
  else:
    diff = m-n
    return (gcd(max(n,diff),min(n,diff)))
```

A. The behaviour depends on the exact values of m and n.
B. The function would still compute gcd correctly.
C. The function would not terminate.

3. Write a python program to find GCD of two numbers. (without using Euclid's algorithm)

4. A positive integer n is said to be perfect if the sum of the factors of n, other than n itself, add up to n. For instance 6 is perfect since the factors of 6 are {1,2,3,6} and 1+2+3=6. Likewise, 28 is perfect because the factors of 28 are {1,2,4,7,14,28} and 1+2+4+7+14=28.

   Write a Python function perfect(n) that takes a positive integer argument and returns True if the integer is perfect, and False otherwise.

   Here are some examples to show how your function should work.

```
>>> perfect(6)
True
>>> perfect(12)
False
```