Assignment 2 (Deadline: 26-Oct-2023)

This is a handwritten assignment due at the start of the next class. There will be no late submissions; in any case, you are expected to join the class on time. You may copy this assignment, but you must write the code in your own handwriting. Begin each program with the main function. I am providing the solution to task 0 for your guidance."

Task 0: Input three numbers. Find if the sum of any two numbers is equal to the third number, print "Y" or "N":

```
a = int(input())
b = int(input())
c = int(input())
if a + b == c or a + c  == b or b + c == a:
      print ("Y")
else:
      print("N")
```

IF, IF-ELSE SECTION

Task 01: Input a, b, & c. Find quadratic roots using formula: $\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ with two exceptions:

- if first parameter (a) is zero, print equation is linear has only one root
- if discriminant is negative, print roots are imaginary
- otherwise print both roots

Task 02: Generate a random number in the range of 1 to 10. Prompt the user to guess the number within three attempts. If the user guesses correctly, print 'Winner'; otherwise, print 'Loser'. If the user loses, also print the generated number.

Task 03: Input four random numbers. Sort these numbers using only one additional variable and conditions. Print the numbers in ascending order with a single print statement at the end.

Task 04: Consider a shopkeeper who sells eggs in packs of six only. Customers must purchase the number of packs based on their requirements. Write a program to input the total number of eggs and print the minimum number of packs required by the customer.

Sample Run:

```
Eggs: 15
Packs: 3
Eggs: 12
Packs: 2
Eggs: 19
Packs: 4
```

Task 05: Input three numbers, and without modifying them, print whether the numbers are in order or not.

Sample Run:

```
Numbers: 23            459         169
Numbers are not in order

Numbers: 823           459         669
Numbers are not in order

Numbers: 123           459         669
Numbers are in order
```

Task 06: Input three numbers and print them in ascending order without using any extra variables or modifying the original variables.

Sample Run:

```
23    459   169
23    169   459

823   459   669
459   669   823
```

Task 07: Input the marks of two students and determine whether they have the same grades using a grading chart. Print 'Same Grades' if their marks fall within the same grading range, and 'Different Grades' if their marks fall within different ranges.

Grading Criteria

85 or more	A
80 - 84	A-
75 - 79	B+
70 - 74	B
65 - 69	B-
61 - 64	C+
58 - 60	C
55 - 57	C-
50 - 54	D
49 or less	F

```
Marks 1: 72
Marks 1: 73
SAME Grades

Marks 1: 73
Marks 1: 83
Different Grades

Marks 1: 50
Marks 1: 53
Same Grades
```

LOOP SECTION

Write next programs using while loop:

Task 08: Write even numbers 2-50 in a single line. Next, write odd numbers 1-49 in next line.

Task 09: Input 5 numbers from user and print their product?

Sample Runs:

```
Enter number: 2
Enter number: 3
Enter number: 1
Enter number: 5
Enter number: 6
Product: 360
```

Task 10: Input 5 numbers from user and print maximum number?

Sample Runs:

```
Enter number: 2
Enter number: 3
Enter number: 1
Enter number: 6
Enter number: 5
Max number: 6
```

Task 11: Run a loop ten times. Within each iteration, generate two random numbers. Check and print whether the first random number is larger or the second random number is larger.

Sample Runs:

```
First: 23
Second: 39
Second number is larger
First: 45
Second: 51
Second number is larger
```

```
First: 23
Second: 3
First number is larger
...
```

Task 12: Execute a loop ten times. Within each iteration, generate three random numbers. In the first line, print the numbers as they are generated. In the next line, print the numbers in ascending order. You can use any method to arrange them.

Sample Runs:

```
23 391 46
23 46 391

358 25 125
25 125 358

35 25 59
25 35 59
...
```

FUNCTION SECTION

Task 13: Define a function to print the geometric sequence with starting number, common ratio, and the count of numbers to print as the parameters. Also, write code to comprehensively test the function.

Task 14: Define a function to return the geometric series (sum of sequence) again with starting number, common ratio, and the count of numbers to print as the parameters. Also, write code to comprehensively test the function.

Task 15: Fermat numbers are defined F_n = (*2 power 2 power n*) + 1 for $n \geq 0$. Define a function to return the Nth Fermat number. Also, write code to comprehensively test the function.

Task 16: Write and test a function that accept letter grades of a student in its 7 parameters. The seven subjects listed below. The function computes and returns the grade point average (GPA) of the student. The formula to compute the GPA is: multiply the points corresponds to each grade with the credit hours of the corresponding subject, then add all the values obtained and divide them with sum of credit hours of the all subject. (Skip the W grade case for the time being).

Sr No	Subject	Credit Hours
1	PF	3
2	PF Lab	1
3	PST	2
4	IICT	2
5	IICT Lab	1
6	ECC	3
7	QT	0.5

Grade	Point
A	5
B	4
C	3
D	2
E	1
F	0
W	--

ARRAY SECTION

Task 17: Input N numbers from user and print them in reverse order.

Task 18: Input N numbers in an array and find minimum and maximum value from those.

Task 19: Input two arrays of N numbers from user and print N products are pairs at the same indices in them.

Task 20: Input N numbers from user and reverse them in same array without using another array. Later print array to test the reversal.