National University
Of Computer and Emerging Sciences

Assignment # 03

Course: EE2003-COAL
Instructor: Safia Fatima
Due Date: See Slate

Task 1:
Write a subroutine **diagonal_sum** which will sum the diagonals of a matrix. Remember, we don't have 2D arrays in Assembly language to deal with matrix so we are going to consider 1D array as 2D array. To deal 1D array as 2D array there is a specific logic to calculate the indexes of 2D array so you have to propose that logic/formula and accordingly calculate indexes and calculate the diagonal sum of the matrix.
From the 1st input matrix the primary diagonal elements are [3,15,7,6], sum is 31 and secondary diagonal [3,3,11,10] sum is 27, so total sum is 58. Similarly, from the 2nd input matrix the primary diagonal elements are [1,5,9], sum is 15 and secondary diagonal [3,5,7] sum is 15 so total sum is 30.

Input: 3,8,12,3,8,15,3,2,2,11,7,3,10,5,9,6
Output: 58
Input: 1,2,3,4,5,6,7,8,9
Output: 30

Note: In all cases, there will be equal number of rows and columns of a matrix.

Task 2:
The Euclid's algorithm (or Euclidean Algorithm) is a method for efficiently finding the greatest common divisor (GCD) of two numbers. The Euclidean algorithm is one of the oldest algorithms in common use. The GCD of two integers X and Y is the largest integer that divides both of X and Y (without leaving a remainder). Euclidean algorithm is based on the principle that the greatest common divisor of two numbers does not change if the larger number is replaced by its difference with the smaller number. The Euclid's Algorithm for calculating GCD of two numbers X and Y can be given as follows:

```
def GCD(X, Y):
     if Y == 0:
          return X
     elif X < Y:
          return GCD(Y, X)
     else :
          return GCD(X-Y, Y)
```

Your task is to convert above recursive function into assembly language code. Write your **subroutine** named as **GCD** and follow the same flow as shown above.

Input: 20, 15
Output: 5
Input: 48, 72
Output: 24

Task3:
Fibonacci sequence of numbers and the associated "Golden Ratio" are manifested in nature and in certain works of art. We observe that many of the natural things follow the Fibonacci sequence. At present Fibonacci numbers plays very important role in coding theory. Fibonacci numbers in different forms are widely applied in constructing security coding and professional photography. Fibonacci number series is the sequence of numbers such that each number is the sum of the two preceding ones starting from 0 and 1. Your task is to write recursive subroutine, for calculating Fibonacci series, which require a number **N** that indicates the required number of terms in the series.

Input: 7
Output: 0, 1, 1, 2, 3, 5, 8
Input: 15
Output: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377

Note: Your logic for subroutine should be recursive. Iterative approach will not be accepted.

Note:

- Make *.asm* file for each task. There are three tasks so you should submit 3 files.
- All these tasks must be done using 16-bit assembly programming concepts taught to you in the class.
- Write a generalized code for all questions because I will be checking the output for different set of inputs.
- Read the tasks carefully. They might seem complicated at first read, but a thorough reading can make them simple and easy.
- Start working on them as soon as possible because you won't be able to complete them at end time.
- **Any plagiarized work will be marked straight 0.**
- Good Luck :)