Assembly language Course (CSW-353)
Instructor: Dr. Karim Emara
Assignment: 1
Due date: **28 Oct 2017**

Ain Shams University
Faculty of Computer
& Information Sciences
3rd year **(Fall 2017)**

General instructions:

1) Submit **only running** code that you have tested before.
2) Put the source code of all questions in **one** file.
3) Mark the start of each question with this **comment**: `;;; Question <number>`
4) Rename the .asm file as *yourSection_yourID_yourName*.asm.
   Example: 1_2015170999_FirstName_LastName
5) Submit **only** the **.asm** file. You have to press "Submit" button not the "Save button" to receive your assignment through coursesites.
6) This assignment is intended for *individual* contribution. Sharing ideas or part of answers is considered plagiarism which is not tolerant.

Marks will be deducted for not following the submission rules (-5/instruction).

Q1. Write an assembly program that reads a 3x3 **DWORD array** and print the summation of its diagonal. You can read the array as 9 signed integers in sequence.

Sample input: 1 2 3 4 5 6 7 8 9

Sample output: 15

Q2. Write an assembly program that reads a number **n** from the user, then calculates and prints the *Triangular Number Sequence* up to the **n**th number. Print only one space among the elements of the sequence. (**DO NOT** hard-code the Triangular Number Sequence in your code.)

Consider a sequence of numbers, in which the first element is 1. The second element of the sequence is the previous element plus the position of the second element (i.e., 1 + 2 = 3). The position of the first element is 1, the second element is 2, and so on. The third element of the sequence is the previous element plus the position of the third element (i.e., 3 + 3 = 6). If we continue this sequence, we get what is called the Triangular Number Sequence, as follows:

1, 3, 6, 10, 15, 21, 28, 36, 45, ...

Sample input: 5
Sample output: 1 3 6 10 15

Q3. Write an assembly program that Transpose a 3 × 3 Matrix.

For example, the 3 × 3 matrix [1, 2, 3, 4, 5, 6, 7, 8, 9] becomes [1, 4, 7, 2, 5, 8, 3, 6, 9]. Solve this problem **without** using shift or rotate instructions or stack instructions (push or pop).

Sample Run:

```
Enter Matrix Elements: 1
                       2
                       3
                       4
                       5
                       6
                       7
                       8
                       9
 Output: 1
         4
         7
         2
         5
         8
         3
         6
         9
```

Q4. Write an assembly program that takes a number **n** from the user and draws a *diamond* shape of asterisks where the number of rows equals **2n – 1**.

Sample Run:

4

Output:

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

Q5. Write an assembly program that takes a number and displays its *multiplication table* (from 1 to 9) separated by commas. **DO NOT** use the MUL instruction.

Sample Run:

- Sample Input: 1
  Sample Output: 1, 2, 3, 4, 5, 6, 7, 8, 9
- Sample input: 3
  Sample Output: 3, 6, 9, 12, 15, 18, 21, 24, 27

Q6. Write an assembly program that takes a binary number as a string from the user and converts it into its *unsigned* decimal value. Maximum length of the binary string is 32. **DO NOT** use **ReadBin** function in the Irvine library.

Hint: You can raise a number to a power using a loop of sum

Sample Run

- String length: 8

  Bin String: 11111111

  Decimal: 255

- String length: 5

  Bin String: 11011

  Decimal: 27

Q7. Write an assembly program that concatenates two source arrays into a destination array. The first array is of type BYTE and the second array is of type WORD. The destination array is of type DWORD where it should hold the values of the first array and the second array concatenated.

Sample Run:

Elements of Array1: 6

Input Array1: 1 4 3 2 5 6

Elements of Array2: 4

Input Array2: 9 7 8 5

Concatenated array: 1 4 3 2 5 6 9 7 8 5

Good Luck

Assembly Language Team 2017