CSC 3210 – Assignment #2
Fall 2021
Due date: 10/14/21 11:59 PM

Objective: Learn memory organization/layout, data transfer concepts and instructions, direct memory access, memory allocation.

Requirements:

1. **(5 points) Implement the following expression in assembly language:**

   `AX = (val3 + 7) – (var2 + val1) + (5/3)*7`

   - Assume that `val1`, `val2`, and `val3` are 16-bit integer variables
   - You need to implement the expression the way it is provided, you cannot do any reduction on the expression while implementing it.
   - Initialize `val1` with 12 (decimal), `val2` with 9 (decimal), and `val3` with 2 (decimal)
   - You are ONLY allowed to use **16-bit registers**.
   - Use ONLY **`mov, add, sub`** instructions whenever needed.
   - Use the debugger to verify your answer.

   - **Submit the following:**
     - Save your source code using your last name, `Lastname1.asm` and upload the `Lastname1.asm`
     - Screenshot (showing the code and register window) of `AX` register contains the correct result.

2. **(5 points) Implement the following expression in assembly language:**

   `CX = –val2 – val1 + (-val1 + val3) + 3`

   - Assume that `val1, val2,` and `val3` are **8-bit integer variables**
   - You need to implement the expression the way it is provided, you cannot do any reduction on the expression while implementing it.
   - Initialize `val1` with 12 (decimal), `val2` with 9 (decimal), and `val3` with 2 (decimal)
   - You are NOT allowed to **update the values** stored in `val1, val2,` and `val3`
   - You are only allowed to use **16-bit registers** to hold intermediate results, whenever needed.
   - Use `mov, add, sub, movzx, movzx,` or `neg` instructions whenever needed.
   - Use the debugger to verify your answer.

   - **Submit the following:**
     - Save your source code using your last name, `Lastname2.asm` and upload the `Lastname2.asm`
     - Screenshot (showing the code and register window) of `CX` register contains the correct result.

3. (3 points) True/False

   (2.1) The instruction, `var BYTE '?'`

   The above instruction declares a variable named `var` and keeps in uninitialized.

   (2.2) The instruction, `var DWORD "ABCD"`

   stores the string 'ABCD' in to variable named `var`

   (2.3) The instruction, `var BYTE "ABCD"`

   stores the characters 'A','B','C','D' in an array of characters named `var`

4. (2 points) Declare a variable:

   ```
   Var1 DWORD 2 DUP (6 DUP ( 3 DUP (?)  ) )
   ```

   What is the total size of the array `Var1` ? Explain your answer.

Note:

- **Submit** your source code by **only** uploading **.ASM file** using **iCollege** in the respective assignment dropbox:
- Lastname1.ASM, Lastname2.ASM
- **Put the following information as Comment header** for .ASM files:

  Student: Full name

  Class: CSC3210

  Assignment#: 2

  Description: This program …….
- Follow the program standards as presented in your book. Pay more attention to code comments and consistent indentation.