CSC 3210 – Assignment #4
Fall 2023

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

Requirements:

1. **Use a loop instruction with indirect addressing to solve the problem.**
   - Do not copy the elements to any other array.
   - Use the LOOP and XCHG instruction.
   - The input array, *inputStr* contains elements: "A", "B", "C", "D", "E", "F", "G", "H".
   - The array's elements after running the program should look like: "G", "H", "E", "F", "C", "D", "A", "B".
     o **Submit the following:**
       ▪ Rename the asm file using your last name as Lastname1.asm
       ▪ Screenshot of the code and memory window showing the content of the variable *inputStr*.

2. **Write an assembly program that does the following:**
   - Define the following value **0506-0307-0408-0102h** in the .data segment using the 64-bit unsigned identifier named qVal.
   - **You can subdivide the qVal value into 4 words – 0506, 0307, 0408, 0102**
   - **Extract these words from qVal using PTR operator.**
   - Find the sum of the words. The sum should be D17h.
   - Store the result in any 16-bit register.
     o **Submit the following:**
       ▪ Rename the asm file using your last name as Lastname2.asm
       ▪ Screenshot of the code and memory window showing the result in a 16-bit register.

3. **Consider the following code:**

```
if (var1 > var2) OR (var3 < var2){
    var1 = var2 + var3;
    var2++;
    var3++;
}
else{
    var1--;
    var2--;
    var3--;
}
```

Here `var1, var2` and `var3` are DWORD variables.
`var1` is initialized with 10 (decimal), `var2` is initialized with 11(decimal) and `var3` is initialized with 12 (decimal).
Translate the following code in assembly code (MASM).
You need to implement the logic of the `if-else statement` with JMP, CMP (compare), JL (jump if less), JA (jump if above) instructions.

Note:
   ▪ **Comment header** for .ASM files:
       Student: Full name
       Class: CSC3210
       Assignment#: 4
       Description: This program ………….
   ▪ Follow the program standards as presented in your book. Pay more attention to code comments and

consistent indentation.
- Create a new project for every question. Do not use one project with multiple .asm files.