

COMSC 260

Fall 2020

Programming Assignment 7

Worth 15 points (1.5% of your grade)

DUE: Tuesday, 10/27/20 by 11:59 P.M. on Canvas

Start by downloading the **260_assign7.asm** file from the Programming Assignment 7 folder on Canvas

NOTE: Your submission for this assignment should be a single **.asm** file and a single **.pdf** file. The following naming convention should be used for naming your files: **firstname_lastname_260_assign7.asm** and **firstname_lastname_260_assign7.pdf**. The pdf file that you submit should contain the screenshots of your sample runs of the program (see below). For example, if your first name is “James” and your last name is “Smith”, then your files should be named James_Smith_260_assign7.asm James_Smith_260_assign7.pdf.

COMMENTS (worth 7.5% of your programming assignment grade): Your program should have at least **ten (10)** different detailed comments explaining the different parts of your program. Each individual comment should be, at a minimum, a short sentence explaining a particular part of your code. You should make each comment as detailed as necessary to fully explain your code. You should also number each of your comments (i.e., comment 1, comment 2, etc.).

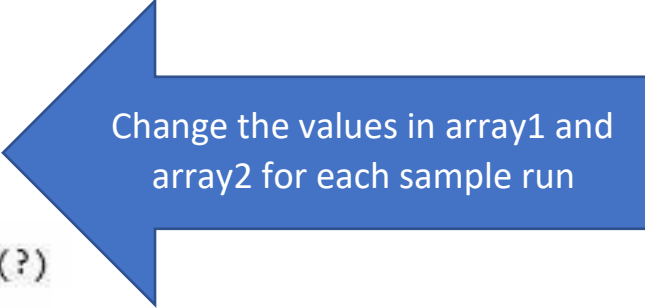
SAMPLE RUNS (worth 7.5% of your programming assignment grade): You should submit screenshots of at least **five (5)** different sample runs of your program. Each sample run needs to use different inputs for the array1 and array2 arrays in the data segment, and your sample runs should **NOT** be the same as the sample run that is used in this write-up for the assignment.

You should also number each of your sample runs (i.e., sample run 1, sample run 2, etc.). All of your sample runs should follow this format – for each individual sample run, screenshot (1) the values used in the array1 and array2 arrays at the beginning of the program and (2) the values in the array1, array2, and array3 arrays at the end of the program,. For example:

(1)

Initial state of the arrays:

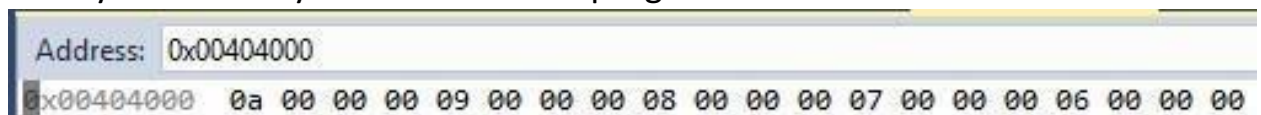
```
.data
    array1 DWORD 1, 2, 3, 4, 5
    array2 DWORD 6, 7, 8, 9, 10
    array3 DWORD LENGTHOF array1 dup(?)
```



Change the values in array1 and array2 for each sample run

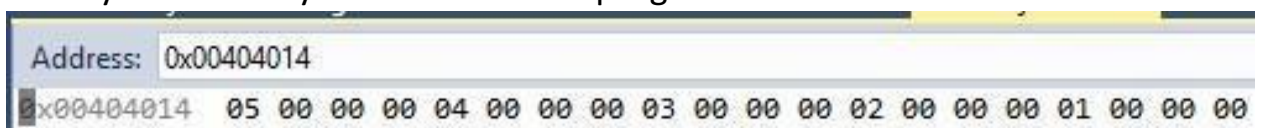
(2)

For &array1 in memory at the end of the program:



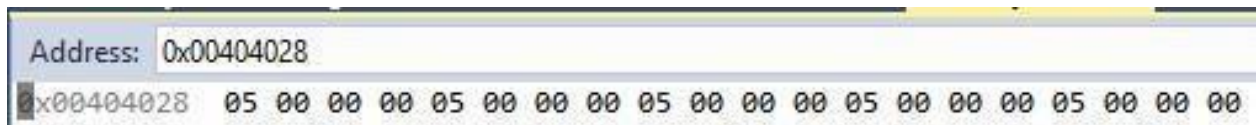
Address: 0x00404000
0x00404000 0a 00 00 00 09 00 00 00 08 00 00 00 07 00 00 00 06 00 00 00

For &array2 in memory at the end of the program:



Address: 0x00404014
0x00404014 05 00 00 00 04 00 00 00 03 00 00 00 02 00 00 00 01 00 00 00

For &array3 in memory at the end of the program:



For this programming assignment you should **ONLY** use these instructions:

- 1) mov (regular/standard move)
- 2) sub

Do NOT use any other instructions besides mov and sub

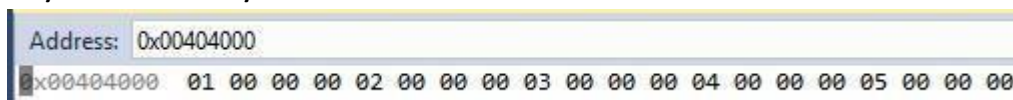
Operands for instructions should be registers or memory locations. An exception is that you can use the immediate value 0 to clear a register.

0 is the ONLY immediate/literal value that should be used in instructions.

You are given the following initial state in memory:

```
.data
    array1 DWORD 1, 2, 3, 4, 5
    array2 DWORD 6, 7, 8, 9, 10
    array3 DWORD LENGTHOF array1 dup(?)
```

&array1 in memory



&array2 in memory

Address: 0x00404014
0x00404014 06 00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0a 00 00 00 00

Watch window:

Watch 1	
Name	Value
array1	0x00000001
array2	0x00000006
array3	0x00000000

PHASE 1:

Phase 1 is approximately twenty (20) instructions

Initially, you have the following arrays:

array1 = [1, 2, 3, 4, 5]

array2 = [6, 7, 8, 9, 10]

You need to swap the two

arrays such that:

array1 = [10, 9, 8, 7, 6]

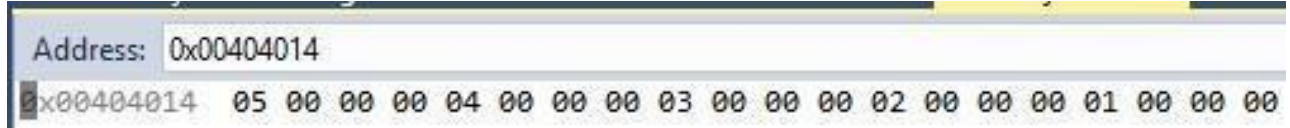
array2 = [5, 4, 3, 2, 1]

After phase 1 is completed, you should have the following:

For &array1 in memory:

Address: 0x00404000
0x00404000 0a 00 00 00 09 00 00 00 08 00 00 00 07 00 00 00 06 00 00 00

For &array2 in memory:



Watch window:

Watch 1	
Name	Value
array1	0x0000000a
array2	0x00000005
array3	0x00000000

PHASE 2:

Phase 2 is approximately fifteen (15) instructions After

completing phase 1, the two arrays are:

array1 = [10, 9, 8, 7, 6]

array2 = [5, 4, 3, 2, 1]

For phase 2, you need to

populate array3 by

subtracting each index in

array2 from the

corresponding index in

array1:

array3[0] = array1[0] -

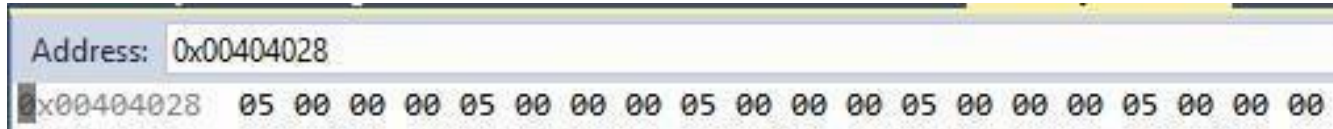
array2[0] = 10 - 5 = 5

array3[1] = array1[1] -

array2[1] = 9 - 4 = 5 etc.

You will end up with array3 = [5, 5, 5, 5, 5]

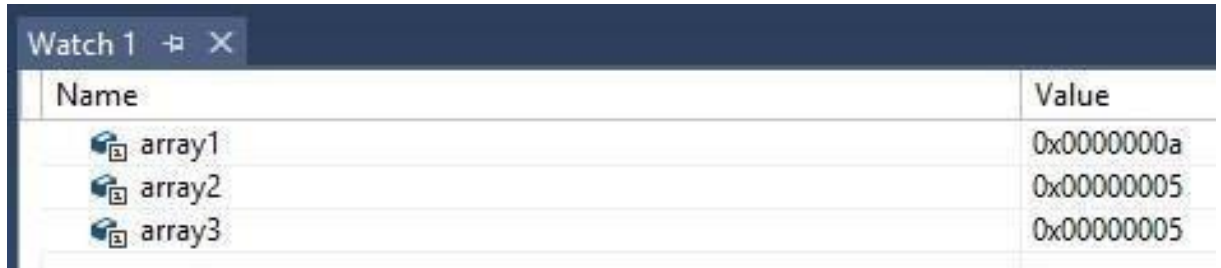
For &array3 in memory:



A memory dump showing the address 0x00404028. The data is displayed in hexadecimal: 05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00. This represents the array [5, 5, 5, 5, 5] in little-endian format.

Address:	0x00404028
0x00404028	05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00

Watch window:



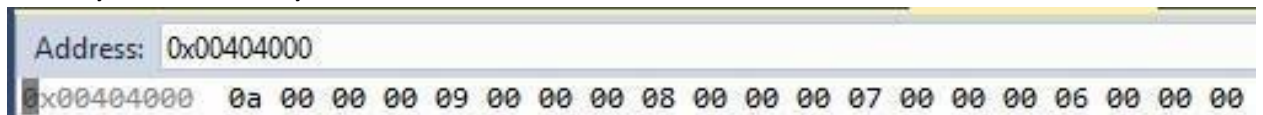
A screenshot of the Watch window showing three variables: array1, array2, and array3. array1 has a value of 0x0000000a, array2 has a value of 0x00000005, and array3 has a value of 0x00000005.

Name	Value
array1	0x0000000a
array2	0x00000005
array3	0x00000005

PROGRAM FINAL OUTPUTS

To recap, after the entire program has finished you should have all of this:

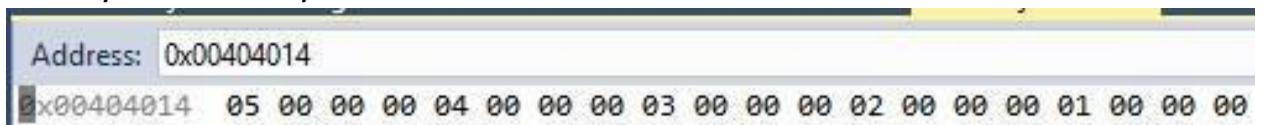
For &array1 in memory



A memory dump showing the address 0x00404000. The data is displayed in hexadecimal: 0a 00 00 00 09 00 00 00 08 00 00 00 07 00 00 00 06 00 00 00. This represents the array [10, 9, 8, 7, 6] in little-endian format.

Address:	0x00404000
0x00404000	0a 00 00 00 09 00 00 00 08 00 00 00 07 00 00 00 06 00 00 00

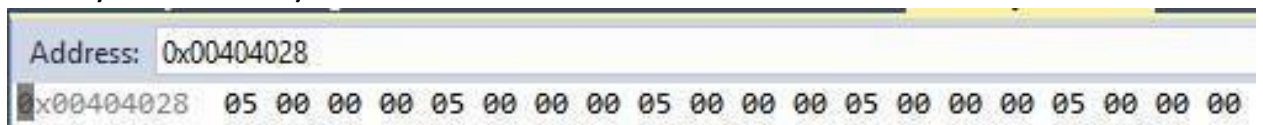
For &array2 in memory



A memory dump showing the address 0x00404014. The data is displayed in hexadecimal: 05 00 00 00 04 00 00 00 03 00 00 00 02 00 00 00 01 00 00 00. This represents the array [5, 4, 3, 2, 1] in little-endian format.

Address:	0x00404014
0x00404014	05 00 00 00 04 00 00 00 03 00 00 00 02 00 00 00 01 00 00 00

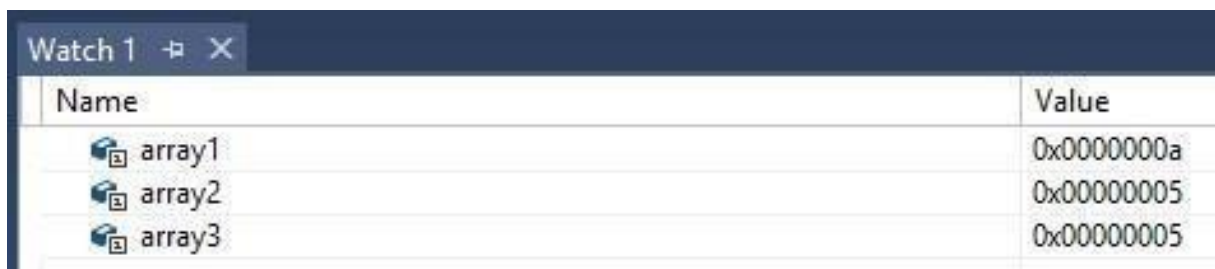
For &array3 in memory:



A memory dump showing the address 0x00404028. The data is displayed in hexadecimal: 05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00. This represents the array [5, 5, 5, 5, 5] in little-endian format.

Address:	0x00404028
0x00404028	05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00 05 00 00 00

Watch window:



A screenshot of the Watch window showing three variables: array1, array2, and array3. array1 has a value of 0x0000000a, array2 has a value of 0x00000005, and array3 has a value of 0x00000005.

Name	Value
array1	0x0000000a
array2	0x00000005
array3	0x00000005