Comp 3350: Computer Organization & Assembly Language
HW # 3: Theme: Data declarations, Small program

All main questions carry equal weight.
(Credit awarded to only those answers with work shown)

1. Explain the two ways of generating a clock for a CPU, as discussed in the class and explain which is preferable and state reasons.
   Ans:

   Simple clock oscillator - Simple clock oscillator is probably the simplest oscillator possible, having only three components. Notice that the gate is a Schmitt inverter. This device has an extremely fast change over between logic states.
   Crystal clock oscillator - Crystal controlled clock oscillator uses three gates from a 74HCT04 IC, and a crystal to provide an accurate frequency of oscillation.
   Crystal clock oscillator is preferable because it is more stable and sufficient.

2. Discuss a synchronous memory read cycle.
   Ans:

   1.Place the address of the value you want to read on the address bus.
   2.Assert (changing the value of) the processor's RD (read) pin.
   3.Wait one clock cycle for the memory chips to respond.
   4.Copy the data from the data bus into the destination operand

3. Declare the following: 3.5 points * 4
   A. An un-initialized data declaration each for a 32-bit signed and unsigned integer
      Ans:

```
// Signed integer
    val1 SDWORD ?
// Unsigned integer
    val2 DWORD ?
```

      data type: 2.5 points
      ?: 1 points

   B. An initialized data declaration for a 32-bit signed integer with the value "9876h" and a 16- bit signed integer with the value "0A4h"
      Ans:

```
// 9876h
    val3 SDWORD 9876h
// 0A4h
    val4 SWORD 0A4h
```

      data type: 2.5 points
      value: 1 point

   C. A null terminated string variable with the value "Computer Architecture".
      Ans:

```
// "Computer Architecture"
    str1 BYTE "Computer Architecture", 0
```

      data type: 2 points
      value: 0.5 points
      terminator: 1 point

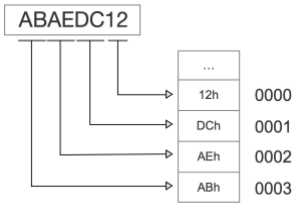
   D. A symbolic constant named "*MinutesinDay*" using the equal-sign directive and assign it an arithmetic expression that calculates the total number of minutes in a Day.
      Ans:

```
// "MinutesinDay"
    MinutesinDay = 24 × 60
```

      Equal-sign: 2.5 points
      value: 1 point

4. Show the order of individual bytes in memory *(lowest to highest)* for the following double word variable *(use little endian order)*: Spring DWORD ABAEDC12h
   Ans:

   3.5 points * 4



5. Show the following using assembler directives: 7 points * 2
   A. How to declare a signed array of five elements and initialize the array with the following values: 91, 81, 71, -61, 51. (You have to choose the right data type which uses the minimum memory space.)
      Ans:

      data type: 5 points
      value: 2 points

      $-2_7 < -61 < 91 < 2_7 - 1$ => SBYTE is the right data type

```
array1 SBYTE 91, 81, 71, -61, 51
```

   B. Using the array created in part A of this question, show how to calculate the number of elements contained and assign that value to a symbolic constant named "*ArrayLength*"
      Ans:

      calculation: 5 points
      value: 2 points

```
// the number of elements = array size / the size of a BYTE
ArrayLength = ($ - array1) / 1
```

6. Why is a string variable declared using the reserved word BYTE as opposed to WORD, DWORD or QWORD?
   Ans:

   Because a string variable is a sequence of one or more characters, each character is stored in memory as a byte.

7. Using the *AddTwo.asm* program from the textbook as a reference, write a program *AddThree.asm* that adds three unsigned word sized integers. Hand write the code. You do not need to assemble/execute.

```
; AddThree.asm - adds three 32-bit integers
; Chapter 3 example

.386
.model flat,stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD

.code
main PROC
     mov eax,5 ; move 5 to the eax register
     add eax,6 ; add 6 to the eax register
     add eax,7 ; add 7 to the eax register

     INVOKE ExitProcess,0
main ENDP
END main
```