**CIS 21JA - Lab 1 Name: \_Tom Ekshtein\_\_\_\_\_\_\_**

*Note: all binary values shown below have a space inserted at every 4 bits to make it easier to read, the space is not part of the actual data value*  
  
1. (1/2pt) The most significant bit in a binary doubleword is bit number \_31\_\_\_\_\_

2. The color value of a pixel in an image is within a range of 0 to 255.

(1/2pt) Is this color value signed or unsigned data? \_\_no\_\_\_\_\_\_\_

(1/2pt) How many bits is needed to represent a data value between 0 to 255? \_\_\_256\_\_\_\_

3. (1pt) The sequence of numbers on the last slide of the module 1 class notes is the short hand notation of how the computer sees a text string.

What does the first value and last value actually look like for the computer? (Hint: what base does a computer work with?)

What does the string look like as a text string for humans?

Since this is a text string I will be using an ASCII chart to convert the hex digits to text.

54 68 65 20 45 6E 64

T h e E n d (20 is a space)

**The End**

Note: To get credit for questions 4-9: *show all work* and read the problem statement carefully so your answer is in the *correct size* for the data.

4. (1.5pt) Convert decimal 28 to binary, and show the result as a byte of data.

2^5 is 32 so we only need to use 2^4 max and will give us only a byte which we are looking for.

2^4=16, 2^3=8, and 2^2=4. 16+8+4 = 28.

Our result: **0001 1100**

5. (1.5pt) Convert decimal -11 to hexadecimal, and show the result as a word of data.  
11 normally in binary is:0000 0000 0000 1011

Its 2’s complement is 1111 1111 1111 0100 + 1 -> 1111 1111 1111 0101

Converting from binary to hex using the chart in the powerpoint: **FFF5**

6. (1.5pt) Convert the *unsigned* hexadecimal value C2 to decimal, and show the result as a byte of data.

Looking at the powerpoint chart, C is 12 and 2 is 2. This means that our total value is (12X16^1)+(2\*16^0)=**194**.

**194** cannot be a byte of data.

7. (1.5pt) Convert the *signed* hexadecimal value C2 to decimal, and show the result as a byte of data.  
  
Using the chart provided in the powerpoint, our binary value for C2 is **1100 0010**. The 2’s complement is 0011 1101 + 1 = 0011 1110 which will allow us to check our work. **1100 0010** to decimal is -2^6+2^2= **-62**. After subtracting **1100 0010** + 0011 1101**,** we get 0 which means our work is correct.

8. (1pt) Do the following binary subtraction in the same way that the CPU would do the subtraction. Show the result of the subtraction as a byte of data  
 1001 1000 – 1010 1000  
  
9. (1/2pt) If X is false, Y is false, and Z is false, show the result of: X xor (Y and Z)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | Y&Z | X V (Y&Z) |
| T | T | T | T |  |
| T | F | F | F |  |
| F | T | T | T |  |
| F | F | F | F |  |
| T | T | T | T |  |
| T | F | F | F |  |
| F | T | T | T |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F** | **F** | **F** | F | F |

Since X is false we need to make sure that Y and Z are True to satisfy the xor condition. However, not only is Y false but Z is as well meaning that this scenario is **False**.

10. (5pts) Follow the instructions at the link "Set up the IDE for MASM" (in module 1) to set up the IDE with an assembly language project that you can use for the rest of the quarter to write your assembly programs.

Then do the following steps to complete the work:

1. At the Solution Explorer window of the IDE, right click on the hello.asm name and select "Remove". This will remove the hello.asm file from the project. **DONE**
2. Download the lab1.asm file. Then right click on the Project name and select "Add Existing Item". Find the lab1.asm file and select it to add it to the project. **DONE**
3. Follow the instruction in the lab1.asm file to modify the code. **DONE**
4. Build and run the program. **DONE**
5. Take a screen shot or a take a picture of your program output screen (the pop up screen with output text that says: This is ... assembly program). Then paste the image in the space below.   
   Make sure you paste an image, not the text output. The image is your proof that you've successfully built and run lab1.asm.

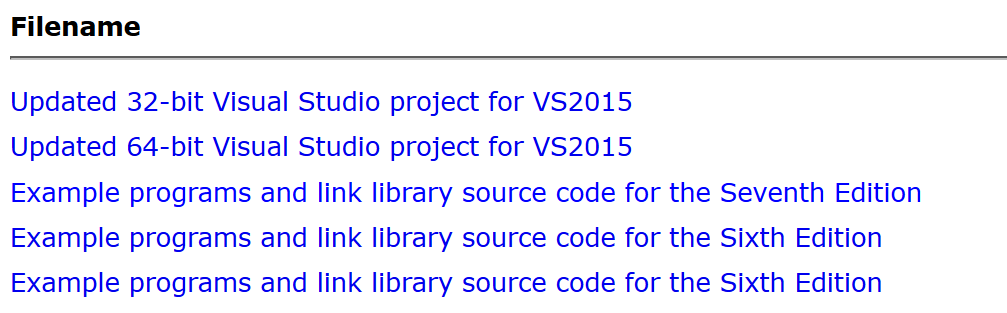
**Steps to set up the IDE and example programs**

A. **Install the free Visual Studio 2015 Express for Desktop IDE from Microsoft**Do the steps in part A only if you don't already have Visual Studio 2013 or 2015 on your computer.

* Click on the Microsoft Visual Studio 2015 for Desktop installer link in Module 1 of the class page to download the executable to your system.
* Double click the installer executable to run it and install Visual Studio IDE with all the standard options when prompted. Depending on your connection speed, it will take minutes to download all the files and install them.

B. **Download the textbook example files and the assembly library files**

* Control+Clickon this [textbook website](http://kipirvine.com/asm/examples/index.htm) link and scroll down to find the link to download the executable to download and install library files

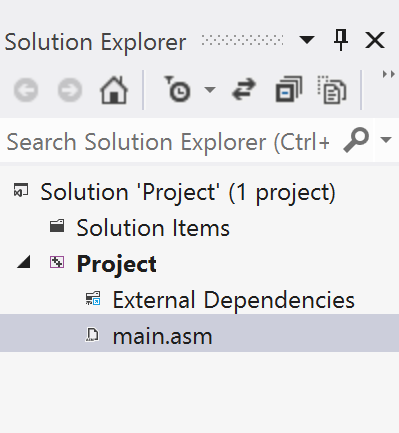


Select this at the download page

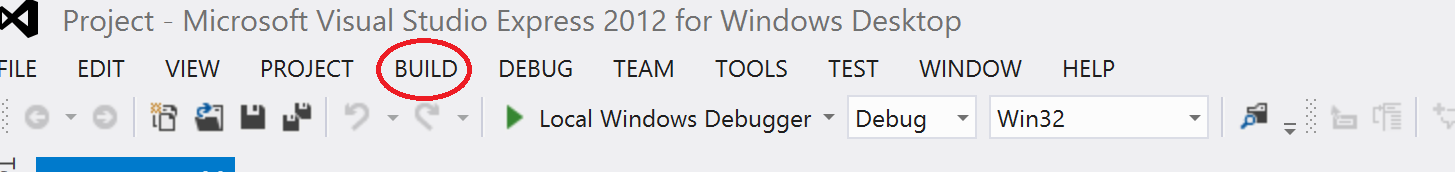
* Click to run the executable and choose all default options to install the sample programs in the textbook and the Irvine library to the folder C:\Irvine on your system.
* When done, go to your C drive and verify that there is an Irvine folder. The Irvine folder should have files such as Irvine32.lib, Irvine32.inc among others.

C. **Run the first assembly program**

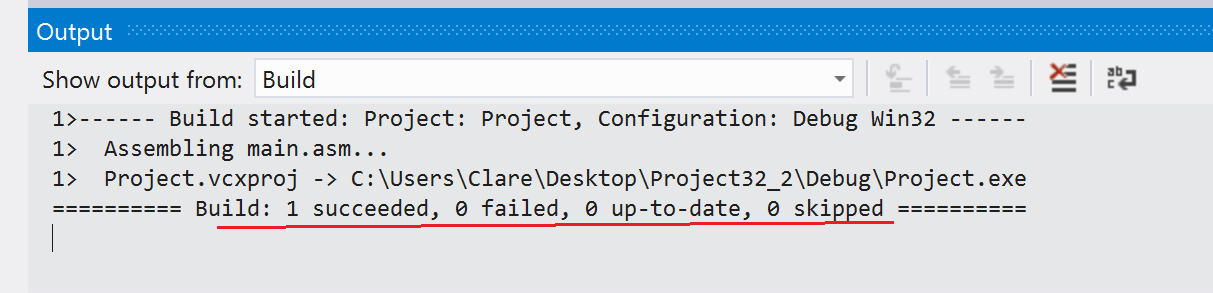
* Download the **Project32.zip** file from Module 1 of the class page on Catalyst to your system.
* Extract the zip file to produce a **Project32** folder. This is the folder that you will use to write and debug assembly programs for the rest of the quarter, so it is recommended that you put the folder on your desktop for easy access.
* Open the Project32 folder and click on the VC++ Project file called **Project** to open the IDE.
* The **Solution Explorer** window will show the **main.asm** program in the project:



* From the **Build** pull down menu,

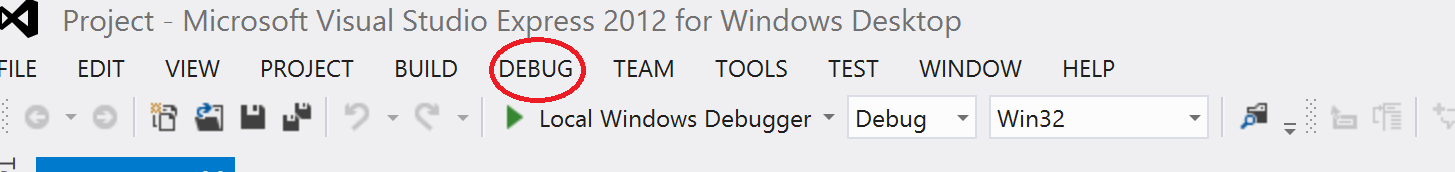


Select **Build Solution** to assemble and link the main.asm program. The build output window should show no syntax or linking error:



Look for this message

* From the **Debug** pull down menu,



select **Start Without Debugging** or use the keyboard short cut Control+F5 to run the executable.

The program output window should appear the classic "Hello world" text output.

By the time you're done with this last step, you will have the basic knowledge of how to build and run an assembly program with the IDE. All the assembly coding done for the rest of the quarter will use this same project and will be run using the same steps.