Project 9: Disassembling C on Windows Part 2 (15 pts. + 10 extra credit)

What You Need

- A Windows machine, real or virtual. I used Windows 7.
- Visual Studio Express, which you installed in a previous project.
- IDA Pro Free, which you installed in a previous project.

Purpose

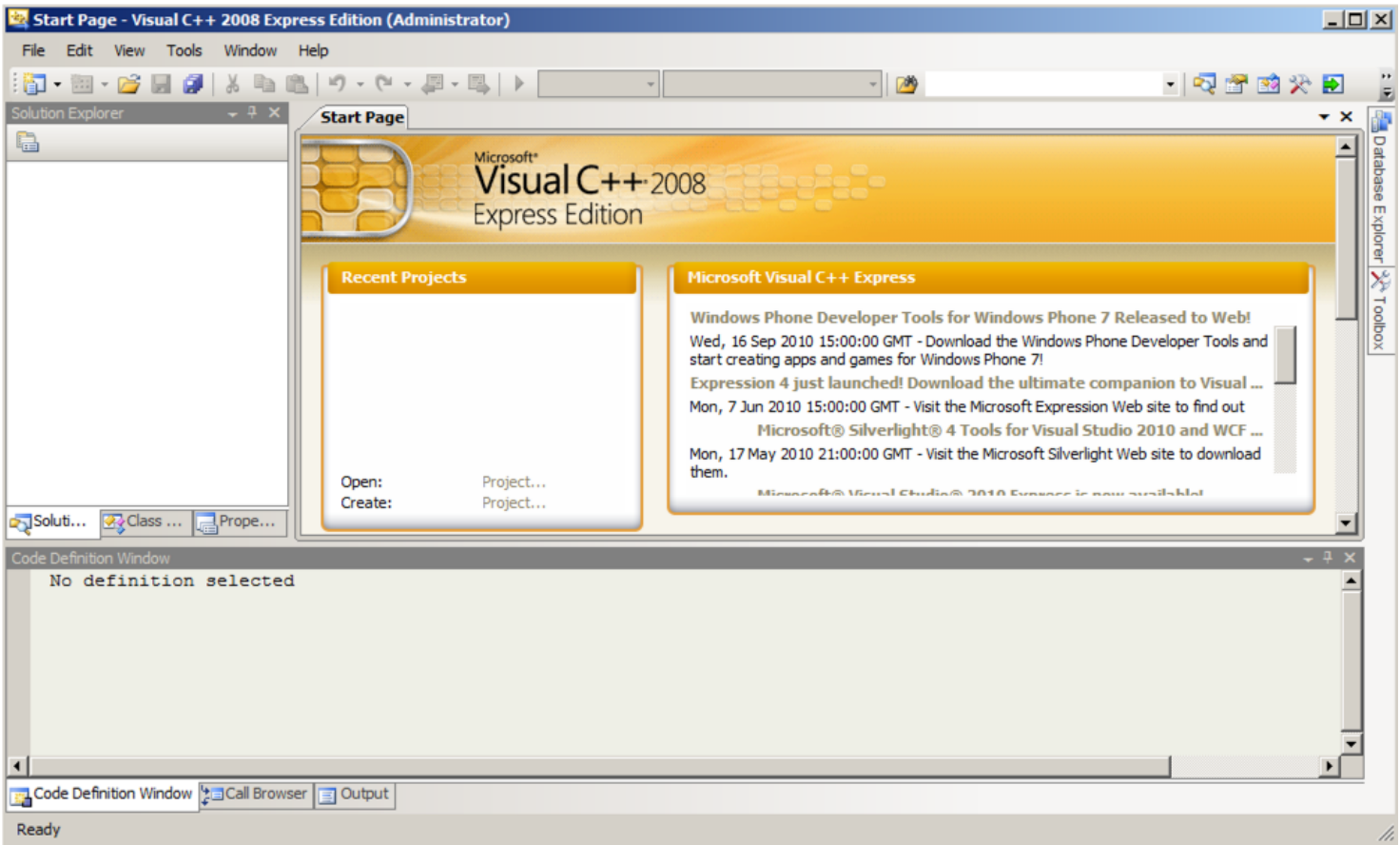
You will write a small C programs using arithmetic statements, compile it, and examine it in the IDA Pro disassembler to learn what it looks like in assembly language.

Launching Visual Studio Express 2008

Click **Start**. Type **VISUAL**

In the search results, click "**Microsoft Visual C++ 2008 Express Edition**"

Visual C++ 2008 Express launches, as shown below:



Making a New C Program

From the "VVisual C++ 2008 Express Edition" menu, click **FILE**, **New**, **Project...**.

In the "New Project" window, on the left, click **Win32**, as shown below.

In the right pane, accept the default selection of "**Win32 Console Application**"

At the bottom of the "New Project" window, type a Name of **YOURNAME-9a**, replacing "YOURNAME" with your own name. Do not use any spaces in the name.

In the "Location" line, click the **Browse** button and navigate to a folder you have permission to save files in, such as your desktop.

Click the "**Select folder**" button.

In the "New Project" window, click **OK**.

A box opens, titled "Welcome to the Win32 Application Wizard".

Click **Next**. In the next screen, accept the default settings and click **Finish**.

A window opens, showing a simple C program.

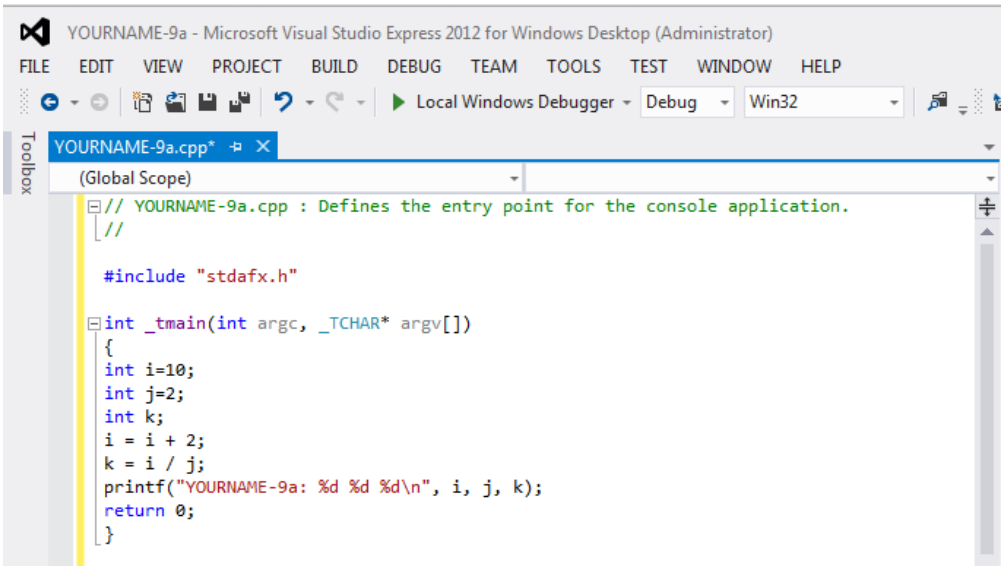
Modify this program to match the code shown in text and the image below.

Do not use the literal string "YOURNAME"--replace it with your own name.

```
// YOURNAME-9a.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
int i=10;
int j=2;
int k;
i = i + 2;
k = i / j;
printf("YOURNAME-9a: %d %d %d\n", i, j, k);
return 0;
}
```



Compiling your Program

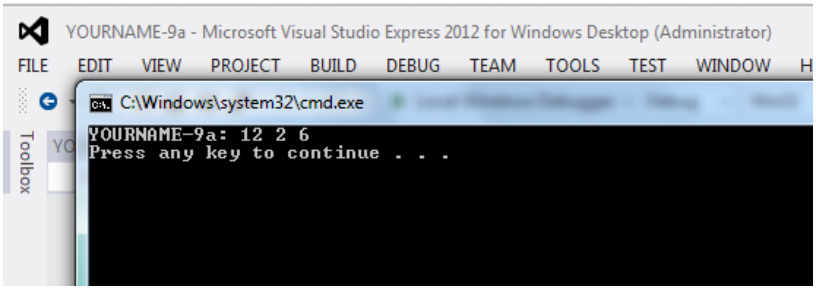
Click **BUILD**, "**Build Solution**".

You should see the message "Build: 1 succeeded" at the bottom of the window. If you see errors, you need to correct them and re-compile the program.

Running your Program

Click **DEBUG**, "**Start Without Debugging**".

A Command Prompt window opens, showing the output of "YOURNAME-9a: 12 2 6", as shown below:



Disassembling the EXE

Click in the Command Prompt window, and press Enter to close it.

Minimize the Visual Studio Express window.

Start IDA Pro Free.

In the "About" box, click **OK**.

Agree to the license.

Close the Help window.

In the "Welcome to IDA!" box, click the **New** button.

In the "New disassembly database" box, double-click "**PE Executable**".

In the "Select PE Executable to disassemble" box, navigate to the folder you used to save your program in Visual Studio Express, probably your desktop.

Double-click the "YOURNAME-9a" folder.

Double-click the **Debug** folder.

Double-click the **YOURNAME-9a.exe** file.

In the "PE Executable file loading Wizard", click **Next**, **Next**, **Finish**.

A box appears, saying this file was linked with debug information.

Click **Yes**

IDA Pro loads the file. As before, the graph mode doesn't show the interesting part of the program.

Expand the **Strings**. Double-click "**YOURNAME-9a %d %d %d\n**".

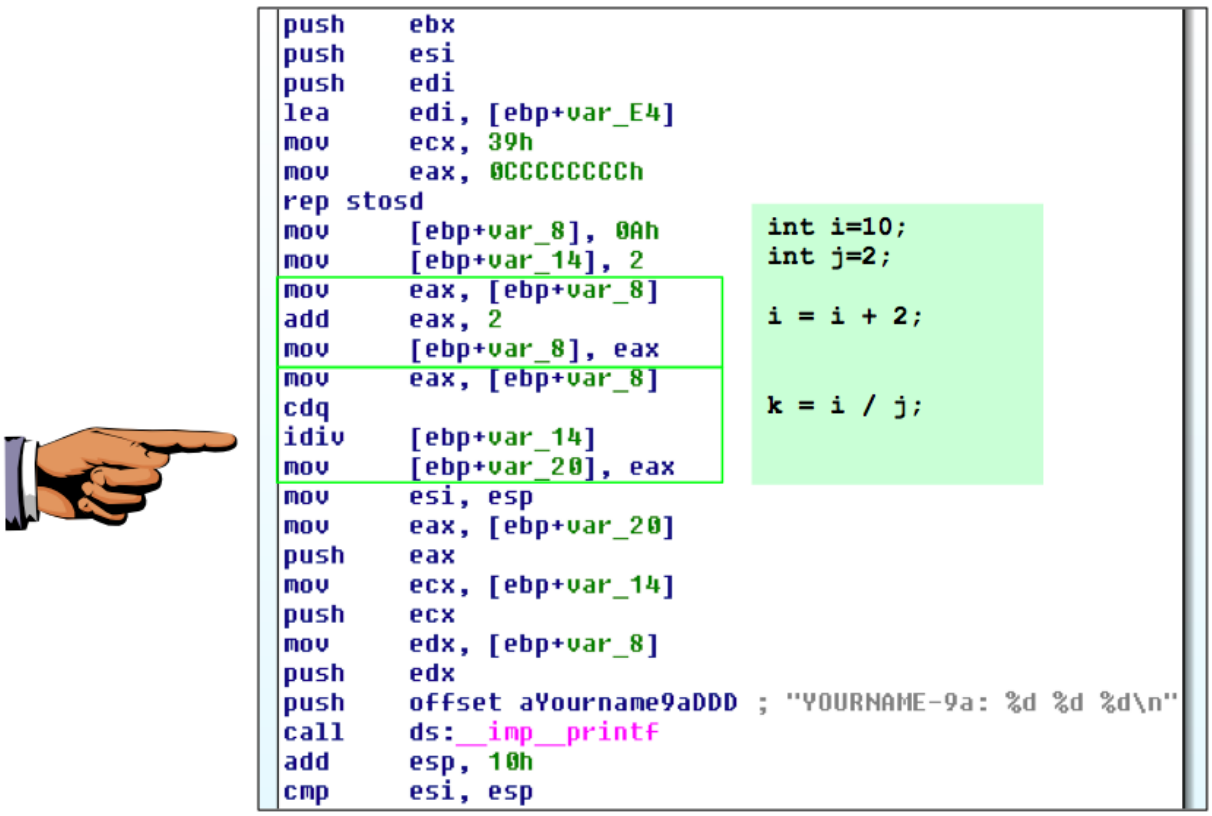
The location containing the string appears.

To the right of "YOURNAME-9a" there is a "DATA XREF" comment. To the right of the "XREF", double-click "**wmain**".

Now the assembly code that performs the task you wrote in C appears, as shown below.

Find the commands listed below, and see how they work. The explanations refer to the C code added to the figure below in the box with green shading.

ASM Code	Explanation	C Code
mov [ebp+var_8], 0Ah	Put the number 10 into a local variable (i)	**int i=10;**
mov [ebp+var_14], 2	Put the number 2 into a local variable (j)	**int j=2;**
mov eax, [ebp+var_8]	Put i into eax	
add eax, 2	Add 2 to eax	**i = i + 2;**
mov [ebp+var_8], eax	Put eax (the result) into a local variable (i)	
mov eax, [ebp+var_8]	Put i into eax	
cdq	Convert double to quad (required for division)	**k = i / j;**
idiv [ebp+var_14]	Divide the value in eax by a local variable (j)	
mov [ebp+var_20], eax	Put eax (the result) into a local variable (k)	



```
push    ebx
push    esi
push    edi
lea     edi, [ebp+var_E4]
mov     ecx, 39h
mov     eax, 0CCCCCCCCh
rep stosd
mov     [ebp+var_8], 0Ah        int i=10;
mov     [ebp+var_14], 2         int j=2;
mov     eax, [ebp+var_8]
add     eax, 2                  i = i + 2;
mov     [ebp+var_8], eax
mov     eax, [ebp+var_8]
cdq                             k = i / j;
idiv    [ebp+var_14]
mov     [ebp+var_20], eax
mov     esi, esp
mov     eax, [ebp+var_20]
push    eax
mov     ecx, [ebp+var_14]
push    ecx
mov     edx, [ebp+var_8]
push    edx
push    offset aYourname9aDDD ; "YOURNAME-9a: %d %d %d\n"
call    ds:__imp__printf
add     esp, 10h
cmp     esi, esp
```

Saving the Screen Image

Make sure you can see the commands listed above, and YOURNAME at the bottom.

On your keyboard, press the PrntScrn key.

Click **Start**, type in **PAINT**, and open Paint.

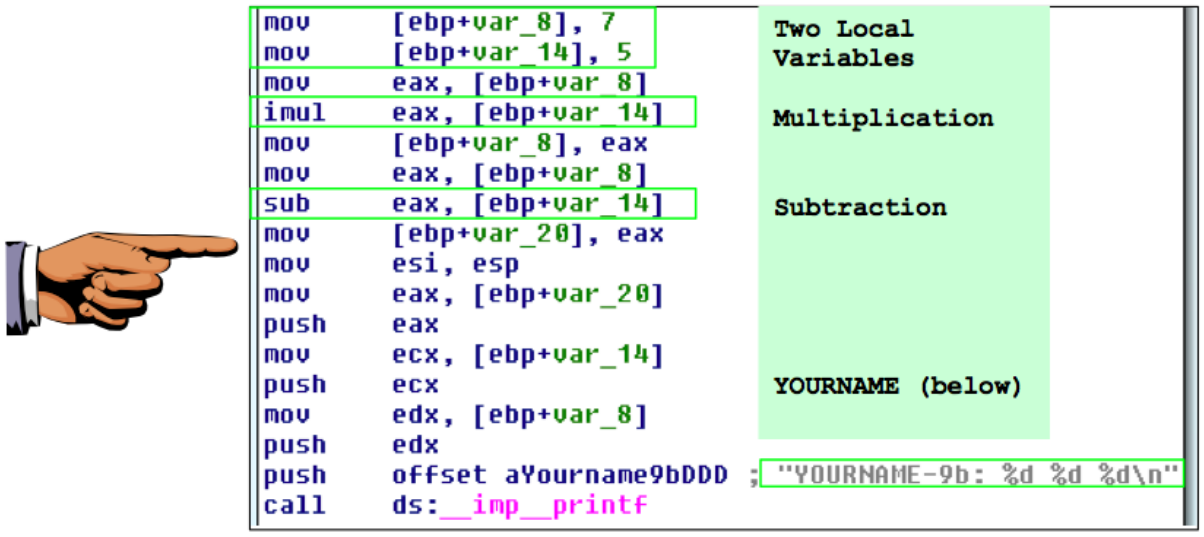
Press **Ctrl+V** to paste in the image of your desktop.

YOU MUST SUBMIT WHOLE-DESKTOP IMAGES TO GET FULL CREDIT.

Save the image with a filename of "**Proj 9a from YOUR NAME**".

CHALLENGE: 10 Pts. Extra Credit

Modify the C program to use multiplication and subtraction, compile it and disassemble it, producing the assembly code shown below.



```
mov     [ebp+var_8], 7          Two Local
mov     [ebp+var_14], 5         Variables
mov     eax, [ebp+var_8]
imul    eax, [ebp+var_14]       Multiplication
mov     [ebp+var_8], eax
mov     eax, [ebp+var_8]
sub     eax, [ebp+var_14]       Subtraction
mov     [ebp+var_20], eax
mov     esi, esp
mov     eax, [ebp+var_20]
push    eax
mov     ecx, [ebp+var_14]
push    ecx                     YOURNAME (below)
mov     edx, [ebp+var_8]
push    edx
push    offset aYourname9bDDD ; "YOURNAME-9b: %d %d %d\n"
call    ds:__imp__printf
```

It must show these features, as labelled in the image above:

- **Two local variables**: two **mov** instructions referencing stack locations such as **[ebp+var_14]**, setting the variables to the values **7** and **5**.
- An **imul** operation
- A **sub** operation
- **YOUR NAME** in the string.

Turning in Your Project

Email the images to: **cnit.126sam@gmail.com** with a subject line of **Proj 9 From Your Name**, replacing Your Name with your own first and last name. Send a Cc to yourself.

Last Modified: 3-20-17