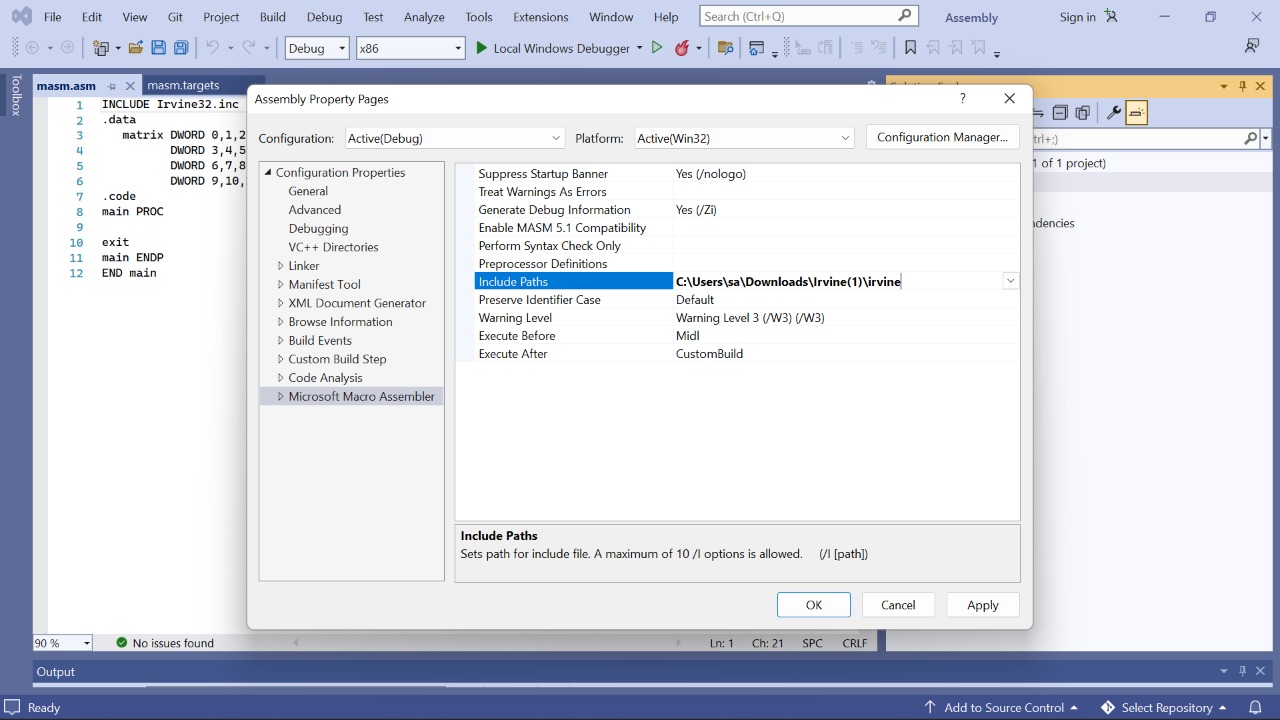
**LAB 4 - 6**

Name: Kulsoom Khurshid

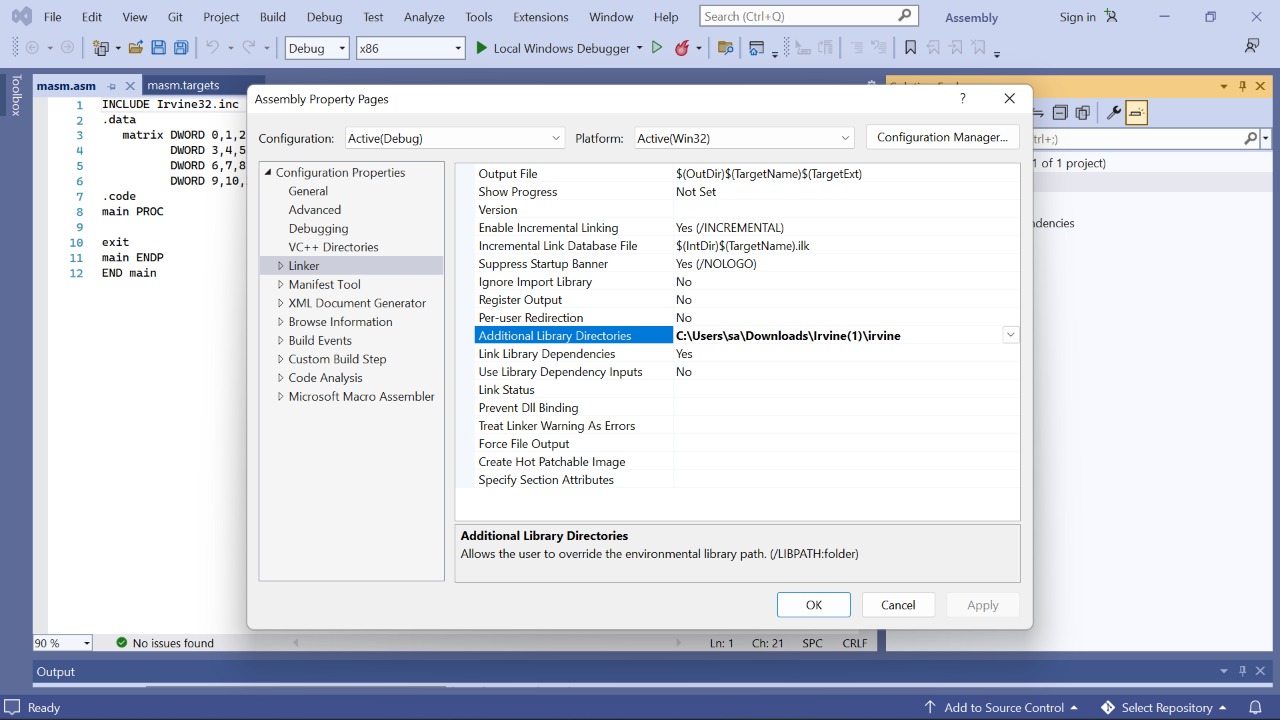
Reg #: Sp20-BCS-044

**LAB # 4:**

1. Assuming that our sample project is still open, select Project Properties from the Project menu. Click the entry named General under Microsoft Macro Assembler. Notice that the Include Paths option has been set to the c:\Irvine directory.

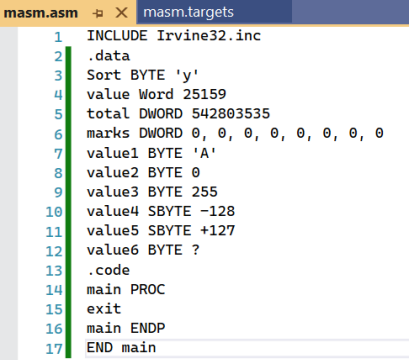


1. Next, select the Listing File entry, also in the Microsoft Macro Assembler group. Find the Linker entry under Configuration Properties. Select the Input entry, and notice that two filenames have been added to the Additional Dependencies entry.

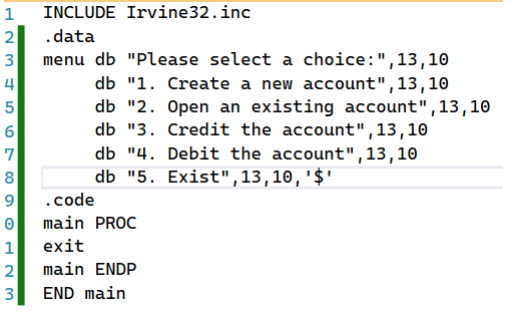


**LAB # 5:**

Home Task 1: Define the following data in Assembly

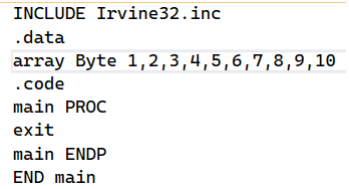


Home Task 2: Create the following menu by declaring a string in Assembly Language

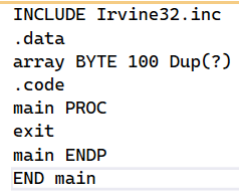


Home Task 3: Create the following arrays in Assembly Language.

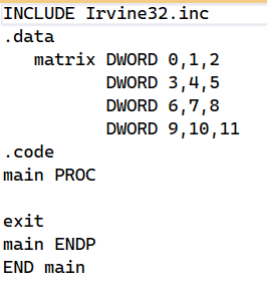
1. Array of ten integers.



1. Array of 100 characters.

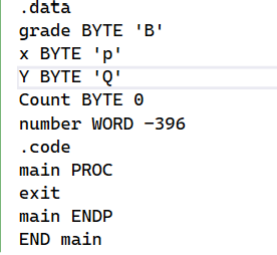


1. Arrays of 4x3 integers



Home Task 4: Implement each of the following declarations in assembly language:

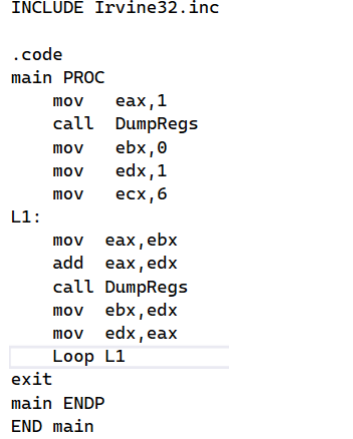
1. char initial;
2. char grade = 'B';
3. char x = 'P', y = 'Q';
4. int amount;
5. int count = 0;
6. int number = -396;



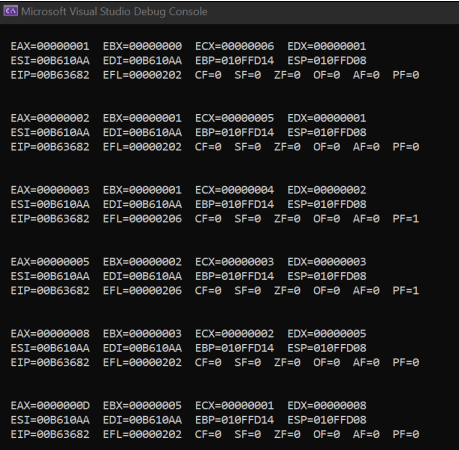
**LAB # 6:**

Home Task 1:

Write a program that uses a loop to calculate the first seven values in the Fibonacci number sequence { 1,1,2,3,5,8,13 }. Place each value in the EAX register and display it with a call DumpRegs statement inside the loop.**CODE:**

****

**OUTPUT:**

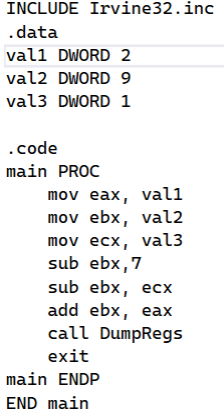
****

Home Task 2:

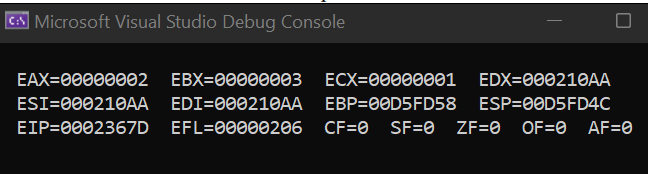
Write a program that implements the following arithmetic expression:

EAX = -val2 + 7 - val3 + val

In comments next to each instruction, write the hexadecimal value of EAX. Insert a call DumpRegs statement at the end of the program.**CODE:**

****

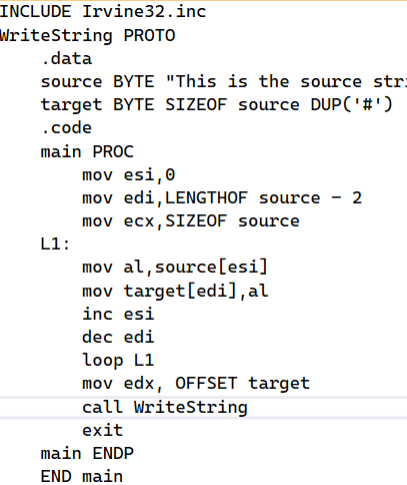
**OUTPUT:**

****

Home Task 3:

Write a program using the LOOP instruction with indirect addressing that copies a string from source to target, reversing the character order in the process.

**CODE:**

****

**OUTPUT:**

