**Lab9**

**Hyunki Lee**

**Problem 1**

a. AL = 0019FF20

b. AL = 0019FF40

c. AX = 0019003B

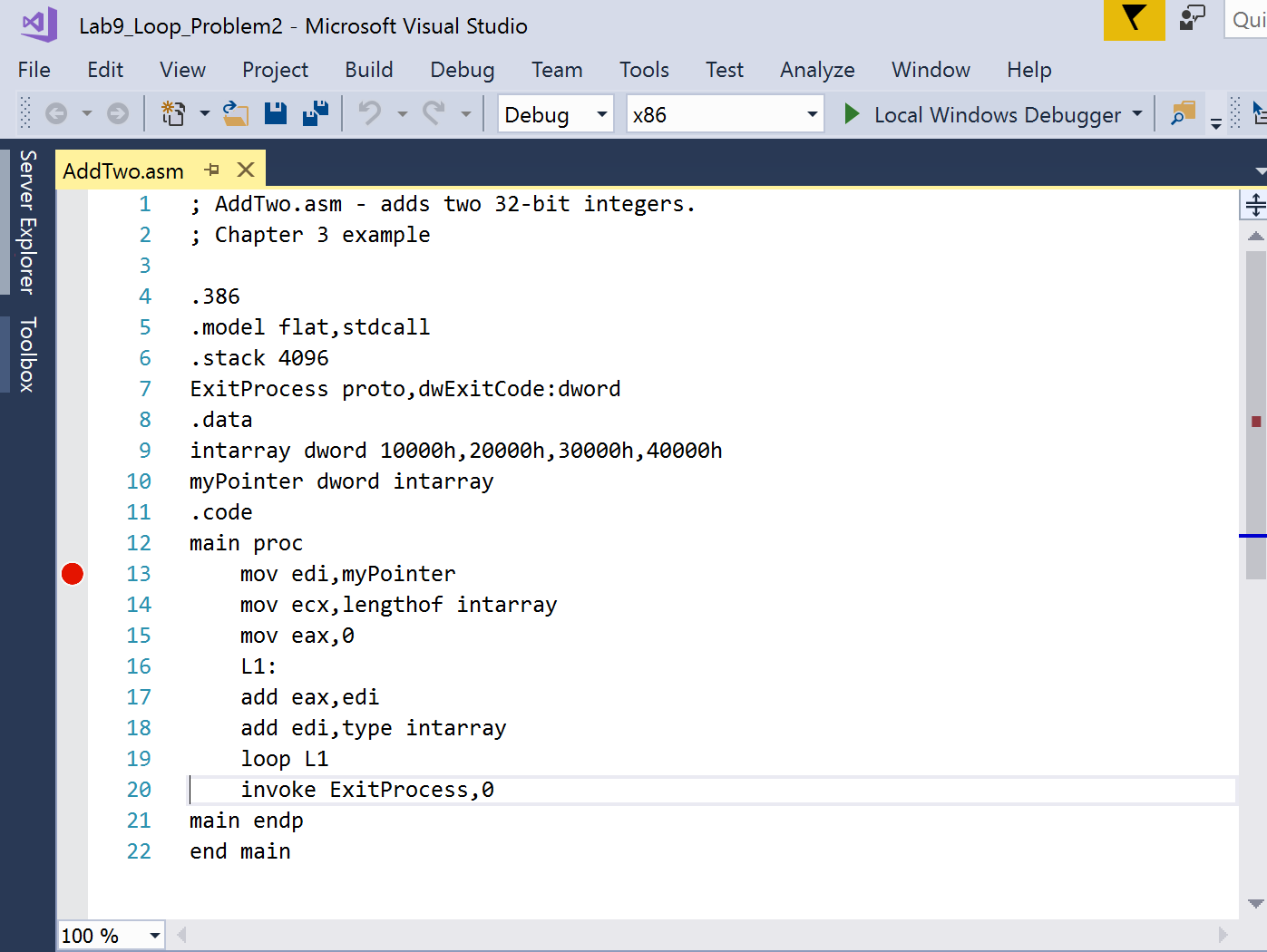
d. EDX = 00000003

e. EDX = 00000003

f. EAX = 00000002

**Problem 2**

Screenshot for pointer



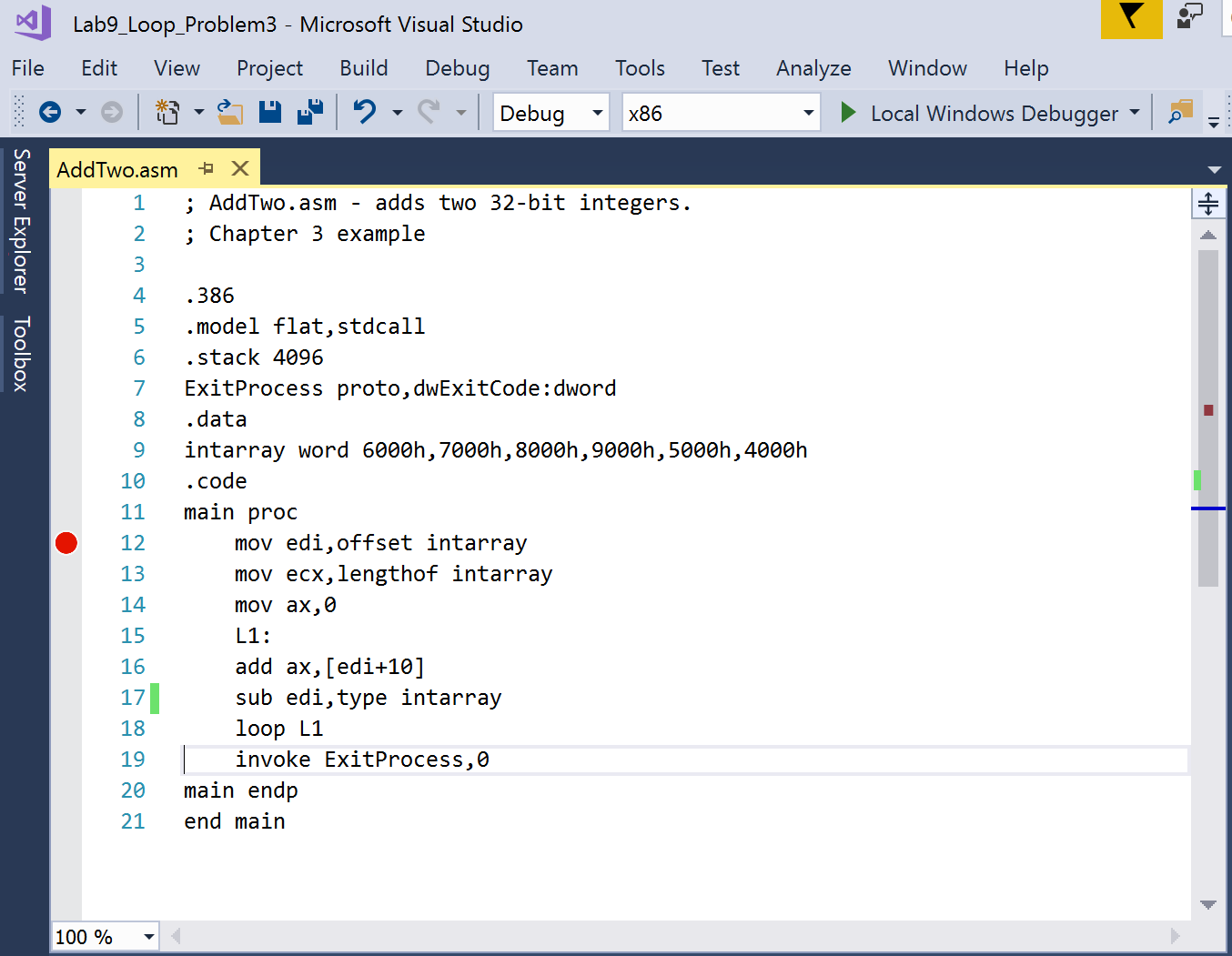
EAX = 000A0000

After removing the square brackets around EDI register

EAX = 01010018

**Problem 3**

Screenshot of Program



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1st loop | add ax,[edi+10]  sub edi,type intarray | CF =0  CF =0 | OV =0  OV =0 | SF =0  SF =0 |
| 2nd loop | add ax,[edi+10]  sub edi,type intarray | CF =0  CF =0 | OV =1  OV =0 | SF =1  SF =0 |
| 3rd loop | add ax,[edi+10]  sub edi,type intarray | CF =1  CF =0 | OV =1  OV =0 | SF =0  SF =0 |
| 4th loop | add ax,[edi+10]  sub edi,type intarray | CF =0  CF =0 | OV =0  OV =0 | SF =1  SF =0 |
| 5th loop | add ax,[edi+10]  sub edi,type intarray | CF =1  CF =0 | OV =0  OV =0 | SF =0  SF =0 |
| 6th loop | add ax,[edi+10]  sub edi,type intarray | CF =0  CF =0 | OV =0  OV =0 | SF =0  SF =0 |

**Problem 4**

1)

Final value of EAX = 0000001C

Tracing ECX

|  |
| --- |
| ECX |
| 0000000A |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 0000000A |
| 00000009 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000009 |
| 00000008 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000008 |
| 00000007 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000007 |
| 00000006 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000006 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000005 |
| 00000004 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000004 |
| 00000003 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000003 |
| 00000002 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000002 |
| 00000001 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000001 |
| 00000000 |

2) Remove mov ecx, temp

The value of ECX becomes FFFFFFFF when processing of debug is going to back into L1.

The program cannot escape from the loops.

3) Using push and pop

I traced the value of ECX.

|  |
| --- |
| ECX |
| 0000000A |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 0000000A |
| 00000009 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000009 |
| 00000008 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000008 |
| 00000007 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000007 |
| 00000006 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000006 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000005 |
| 00000004 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000004 |
| 00000003 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000003 |
| 00000002 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000002 |
| 00000001 |
| 00000005 |
| 00000004 |
| 00000003 |
| 00000002 |
| 00000001 |
| 00000000 |
| 00000001 |
| 00000000 |

I got the same result that is the original program’s output. The red color numbers are the value that is pushed and popped.