Lab9

Hyunki Lee

Problem 1

a. AL = 0019FF20

b. AL = 0019FF40

c. AX = 0019003B

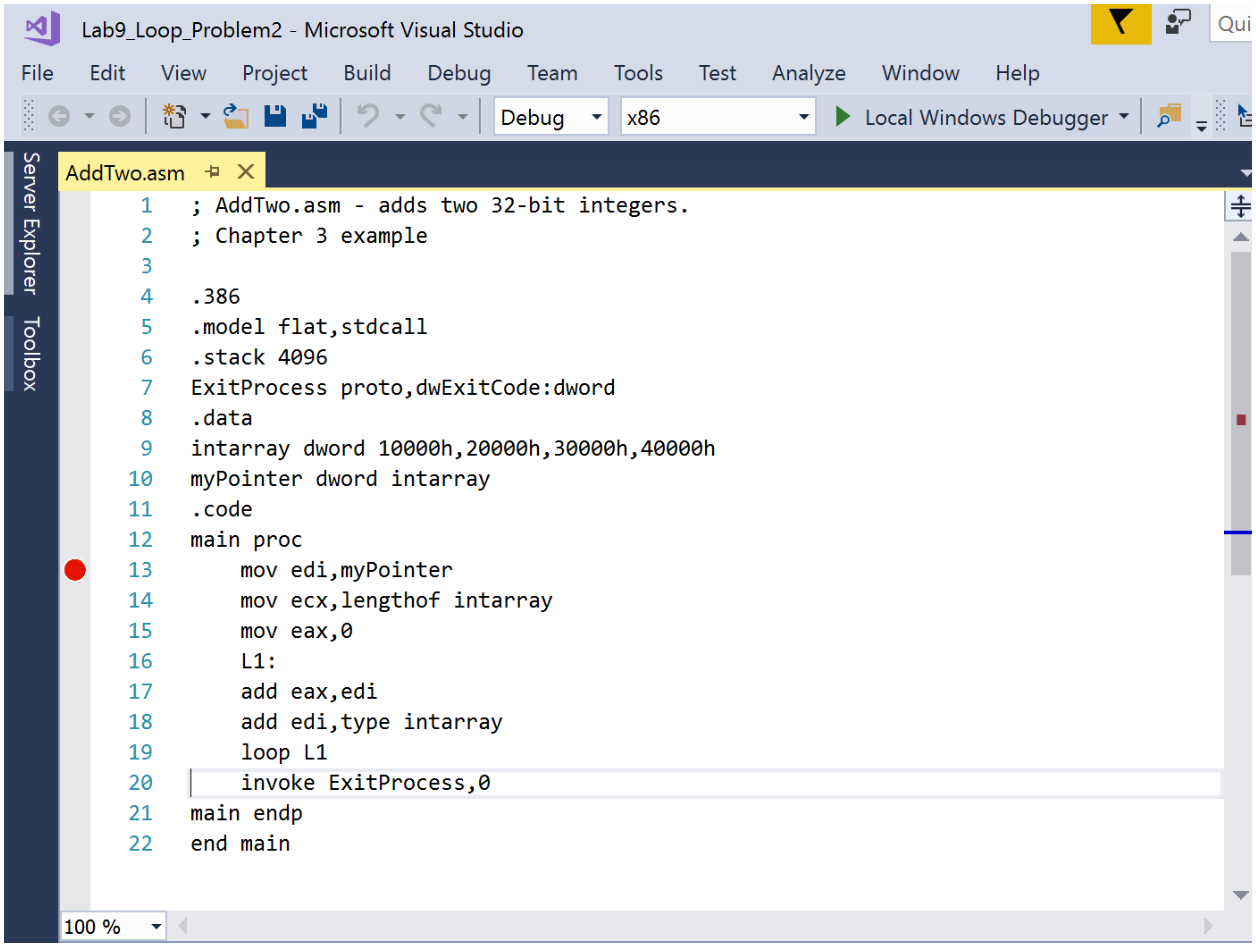
d. EDX = 00000003

e. EDX = 00000003

f. EAX = 00000002

Problem 2

Screenshot for pointer


Lab9_Loop_Problem2 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
Server Explorer
Toolbox
AddTwo.asm
100 %

```
; AddTwo.asm - adds two 32-bit integers.
; Chapter 3 example

.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword
.data
intarray dword 10000h,20000h,30000h,40000h
myPointer dword intarray
.code
main proc
    mov edi,myPointer
    mov ecx,lengthof intarray
    mov eax,0
    L1:
    add eax,edi
    add edi,type intarray
    loop L1
    invoke ExitProcess,0
main endp
end main
```

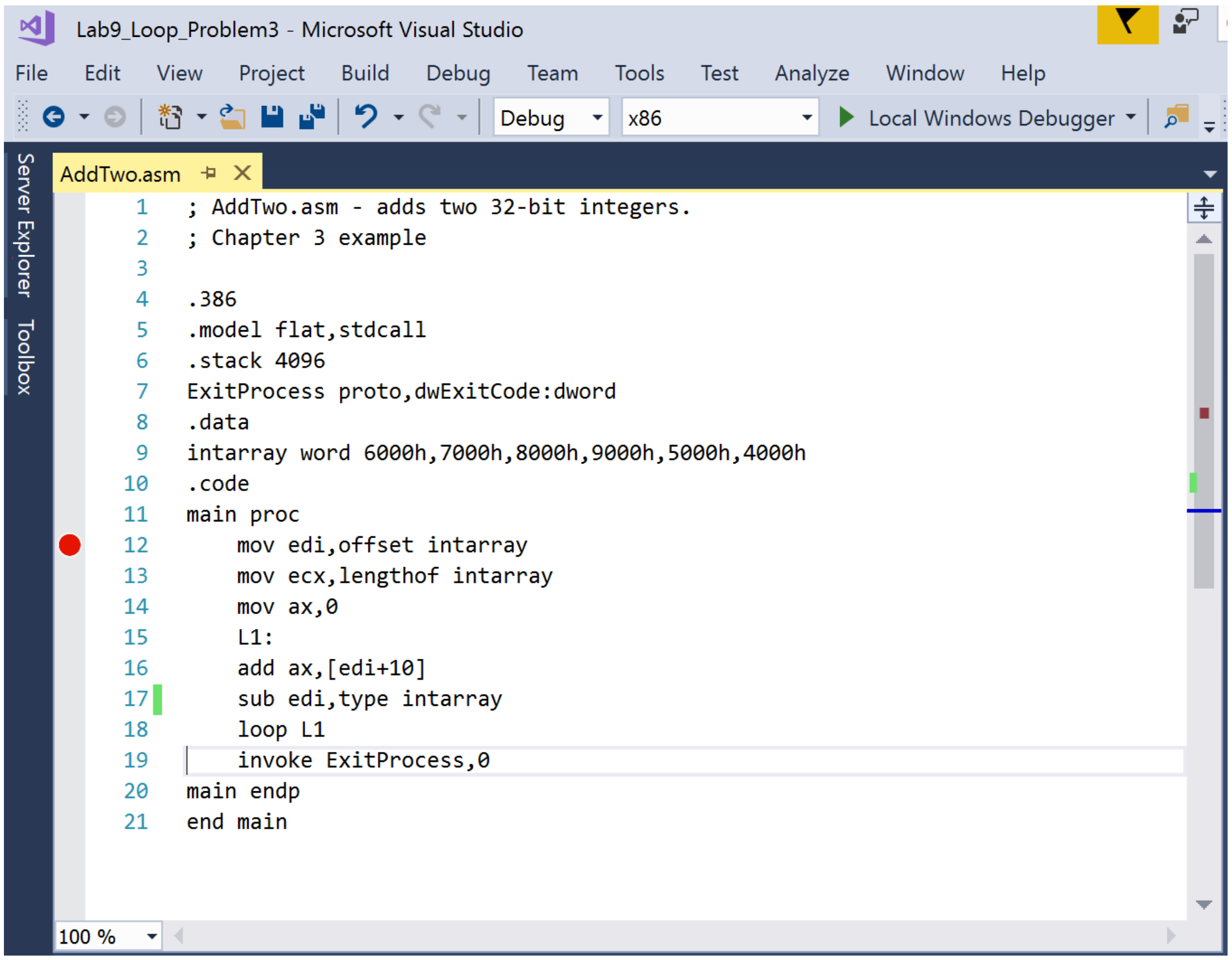
EAX = 000A0000

After removing the square brackets around EDI register

EAX = 01010018

Problem 3

Screenshot of Program


Lab9_Loop_Problem3 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Local Windows Debugger
AddTwo.asm
Server Explorer
Toolbox
100 %

```
; AddTwo.asm - adds two 32-bit integers.
; Chapter 3 example

.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword
.data
intarray word 6000h,7000h,8000h,9000h,5000h,4000h
.code
main proc
    mov edi,offset intarray
    mov ecx,lengthof intarray
    mov ax,0
    L1:
    add ax,[edi+10]
    sub edi,type intarray
    loop L1
    invoke ExitProcess,0
main endp
end main
```

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.