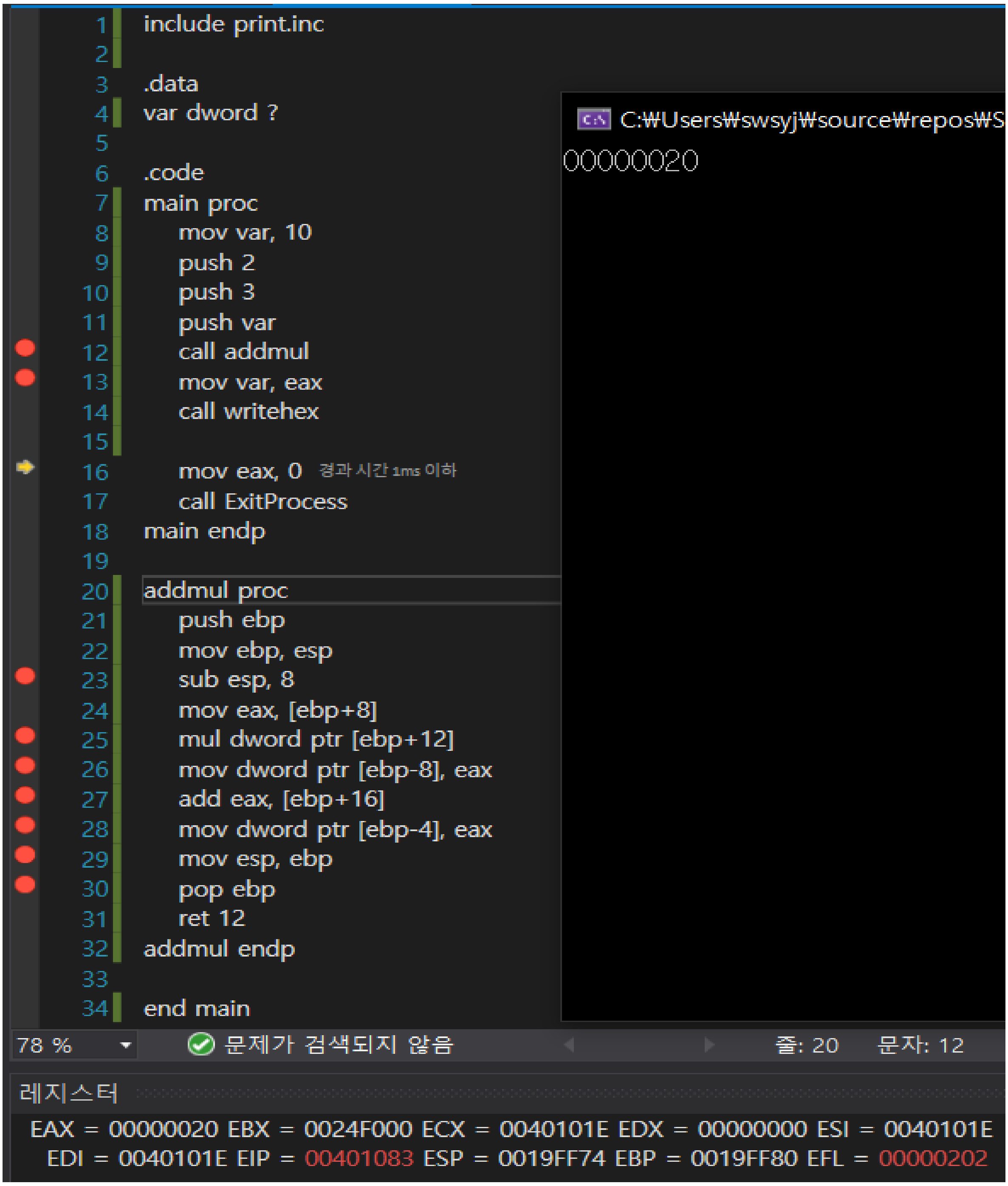
<  과제  6  >

2017253041_홍성우

1.



```
include print.inc

.data
var dword ?

.code
main proc
    mov var, 10
    push 2
    push 3
    push var
    call addmul
    mov var, eax
    call writehex

    mov eax, 0
    call ExitProcess
main endp

addmul proc
    push ebp
    mov ebp, esp
    sub esp, 8
    mov eax, [ebp+8]
    mul dword ptr [ebp+12]
    mov dword ptr [ebp-8], eax
    add eax, [ebp+16]
    mov dword ptr [ebp-4], eax
    mov esp, ebp
    pop ebp
    ret 12
addmul endp

end main
```

C:₩Users₩swsyj₩source₩repos₩S

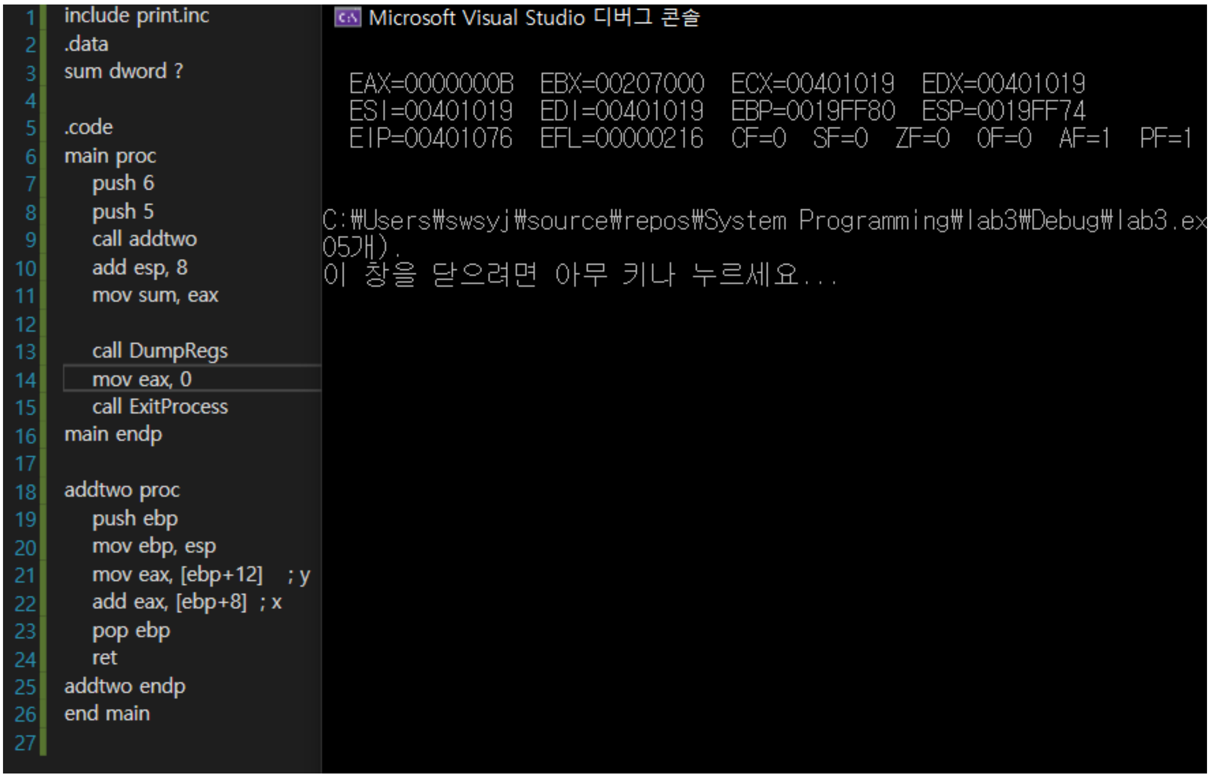
```
00000020
```

```
EAX = 00000020 EBX = 0024F000 ECX = 0040101E EDX = 00000000 ESI = 0040101E
  EDI = 0040101E EIP = 00401083 ESP = 0019FF74 EBP = 0019FF80 EFL = 00000202
```

-전역변수 var를 ?로 설정해주고 C언어 프로그램에 따라 10을 대입하고 차례대로 push해준다. addmul을 call해주고 stack parameter와 local variable에 맞게 분류해서 C언어 프로그램대로 코딩해준다. 마지막엔 stack parameter를 callee에서 ret 12로 cleanup 해주고 ret을 해서 eax값을 var에 넣어주고 call writehex로 eax값을 출력해준다.

2.

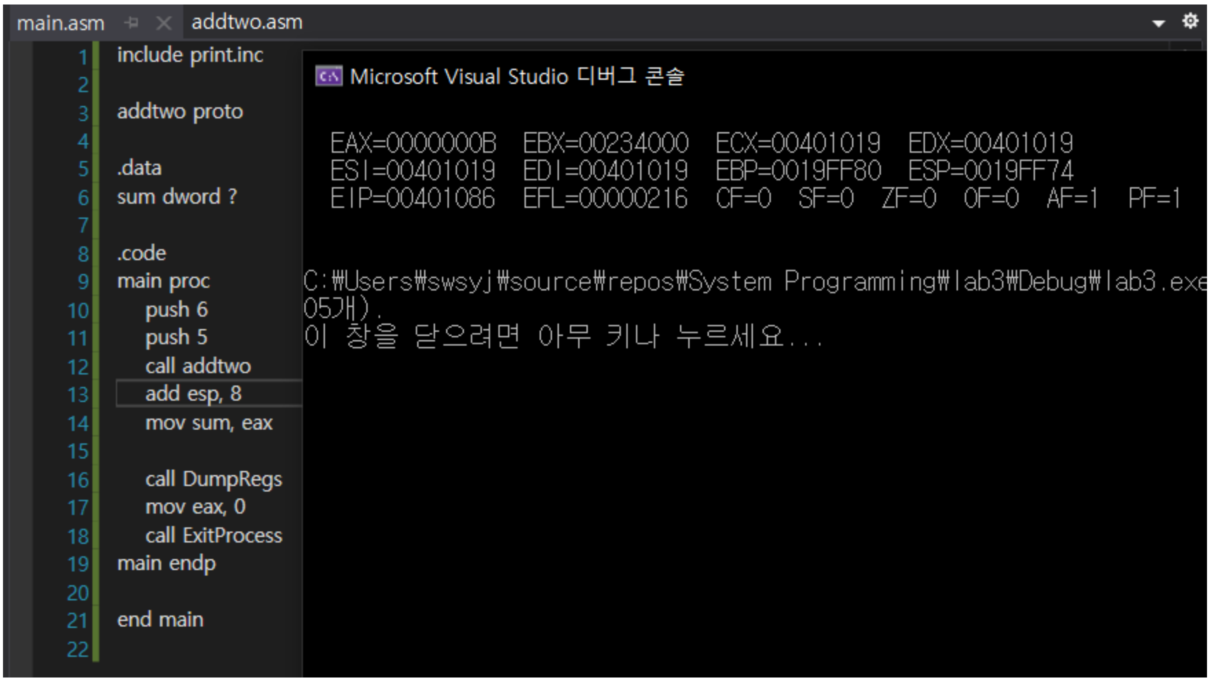
(1)



```
include print.inc
.data
sum dword ?

.code
main proc
    push 6
    push 5
    call addtwo
    add esp, 8
    mov sum, eax

    call DumpRegs
    mov eax, 0
    call ExitProcess
main endp

addtwo proc
    push ebp
    mov ebp, esp
    mov eax, [ebp+12]    ; y
    add eax, [ebp+8]  ; x
    pop ebp
    ret
addtwo endp
end main
```

```
 EAX=0000000B  EBX=00207000  ECX=00401019  EDX=00401019
 ESI=00401019  EDI=00401019  EBP=0019FF80  ESP=0019FF74
 EIP=00401076  EFL=00000216  CF=0  SF=0  ZF=0  OF=0  AF=1  PF=1

C:\Users\swsyj\source\repos\System Programming\lab3\Debug\lab3.ex
05개).
이 창을 닫으려면 아무 키나 누르세요...
```

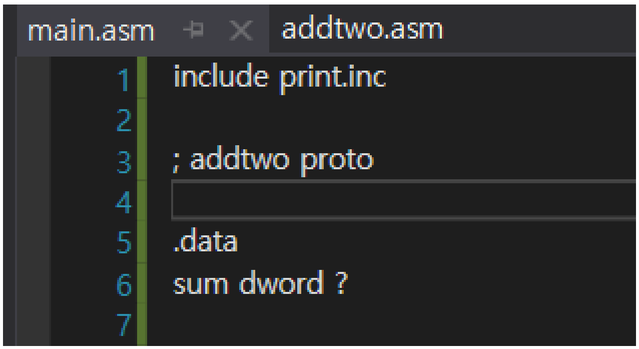
(2)



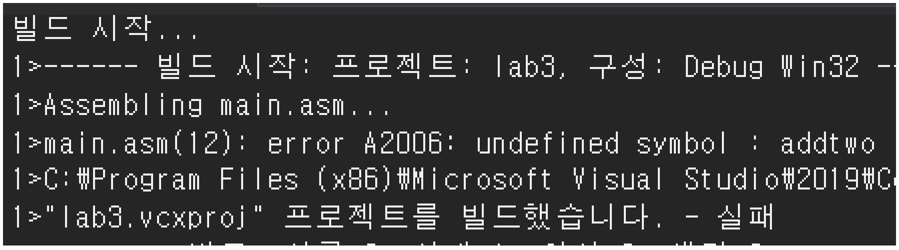
```
include print.inc

addtwo proto

.data
sum dword ?

.code
main proc
    push 6
    push 5
    call addtwo
    add esp, 8
    mov sum, eax

    call DumpRegs
    mov eax, 0
    call ExitProcess
main endp

end main
```

```
 EAX=0000000B  EBX=00234000  ECX=00401019  EDX=00401019
 ESI=00401019  EDI=00401019  EBP=0019FF80  ESP=0019FF74
 EIP=00401086  EFL=00000216  CF=0  SF=0  ZF=0  OF=0  AF=1  PF=1

C:\Users\swsyj\source\repos\System Programming\lab3\Debug\lab3.exe
05개).
이 창을 닫으려면 아무 키나 누르세요...
```

(3)

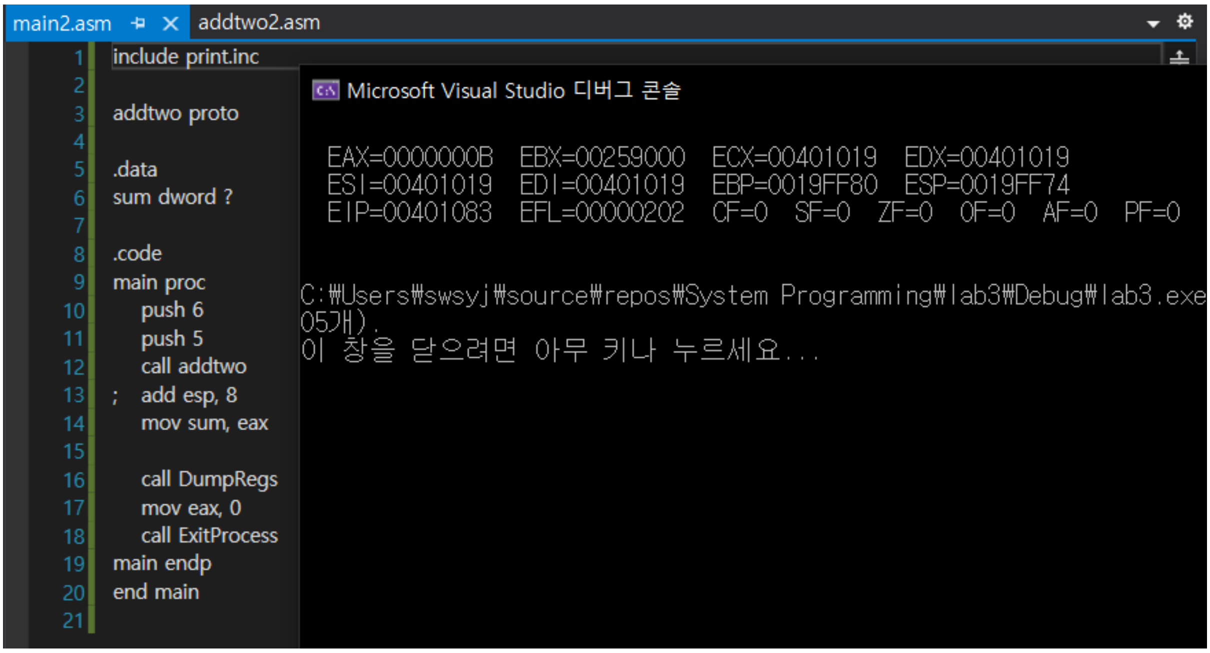


```
include print.inc

; addtwo proto

.data
sum dword ?

```



```
빌드 시작...
1>------ 빌드 시작: 프로젝트: lab3, 구성: Debug Win32 --
1>Assembling main.asm...
1>main.asm(12): error A2006: undefined symbol : addtwo
1>C:₩Program Files (x86)₩Microsoft Visual Studio₩2019₩C
1>"lab3.vcxproj" 프로젝트를 빌드했습니다. - 실패
```

(4)

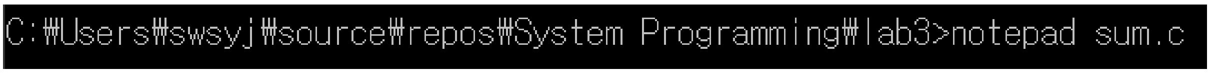


```
include print.inc

addtwo proto

.data
sum dword ?

.code
main proc
    push 6
    push 5
    call addtwo
;   add esp, 8
    mov sum, eax

    call DumpRegs
    mov eax, 0
    call ExitProcess
main endp
end main

```

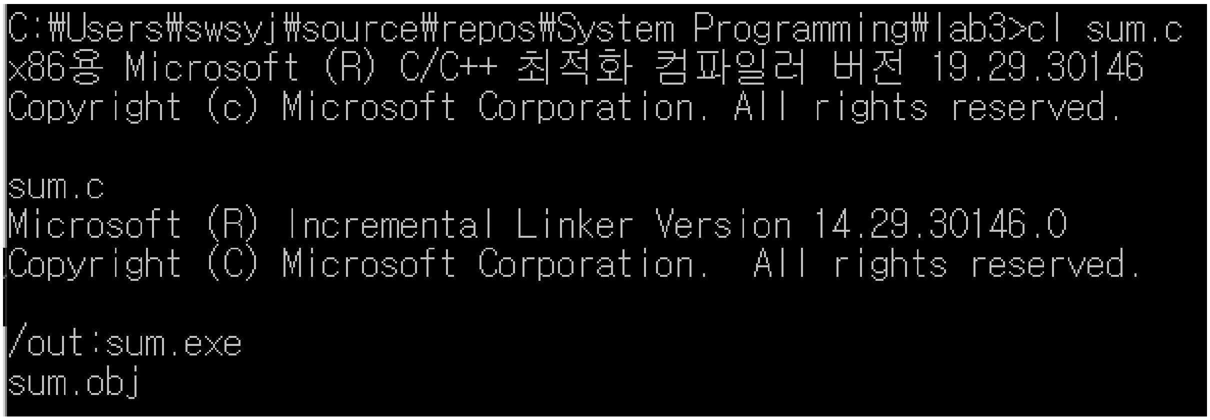
```
  EAX=0000000B  EBX=00259000  ECX=00401019  EDX=00401019
  ESI=00401019  EDI=00401019  EBP=0019FF80  ESP=0019FF74
  EIP=00401083  EFL=00000202  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=0


C:₩Users₩swsyj₩source₩repos₩System Programming₩lab3₩Debug₩lab3.exe
05개).
이 창을 닫으려면 아무 키나 누르세요...
```

-앞에서 작성한 프로그램과 다른 점은 stack parameter를 addtwo.asm에서는 caller가 add esp, 8로 clean up 해줬지만, addtwo2.asm에서는 callee가 ret 8로 clean up 해줬다.

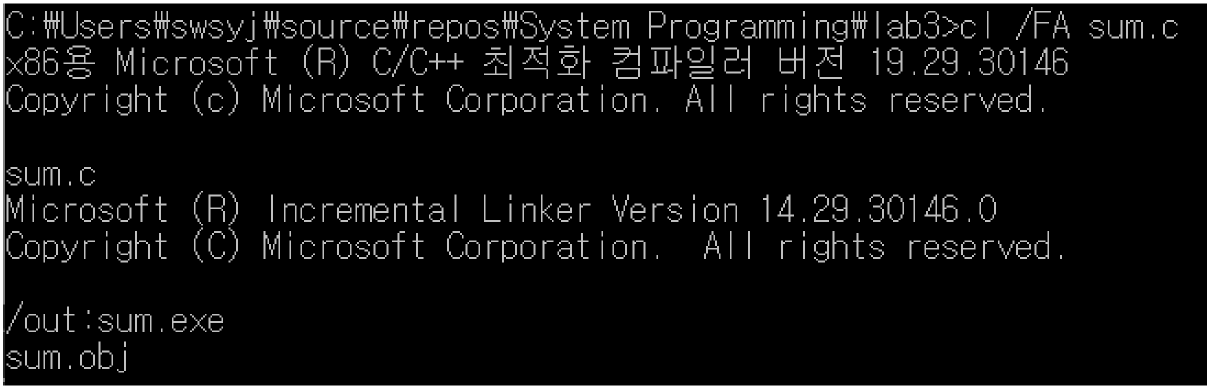
3.



```
C:₩Users₩swsyj₩source₩repos₩System Programming₩lab3>notepad sum.c
```

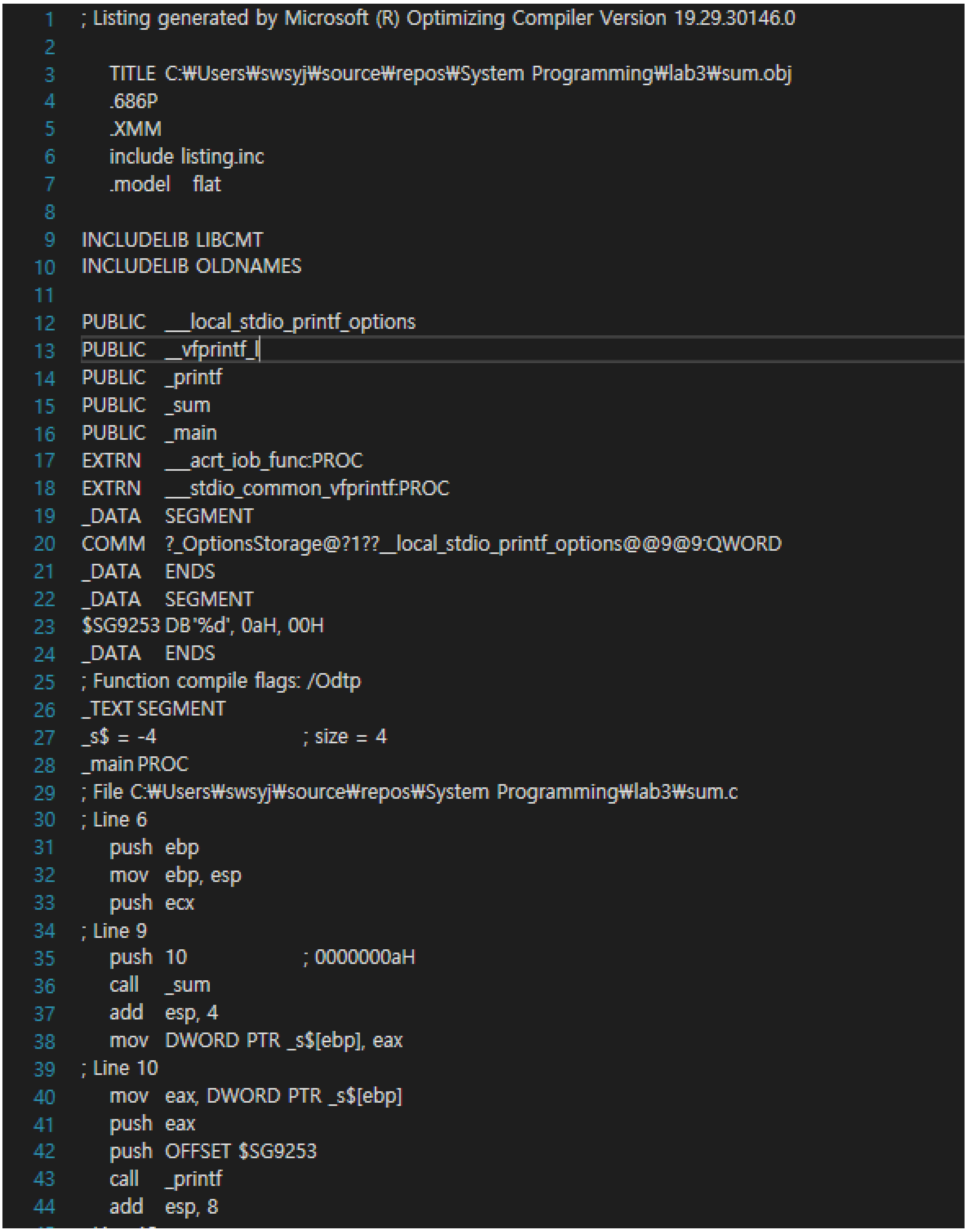


```
C:₩Users₩swsyj₩source₩repos₩System Programming₩lab3>cl sum.c
x86용 Microsoft (R) C/C++ 최적화 컴파일러 버전 19.29.30146
Copyright (c) Microsoft Corporation. All rights reserved.

sum.c
Microsoft (R) Incremental Linker Version 14.29.30146.0
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:sum.exe
sum.obj
```

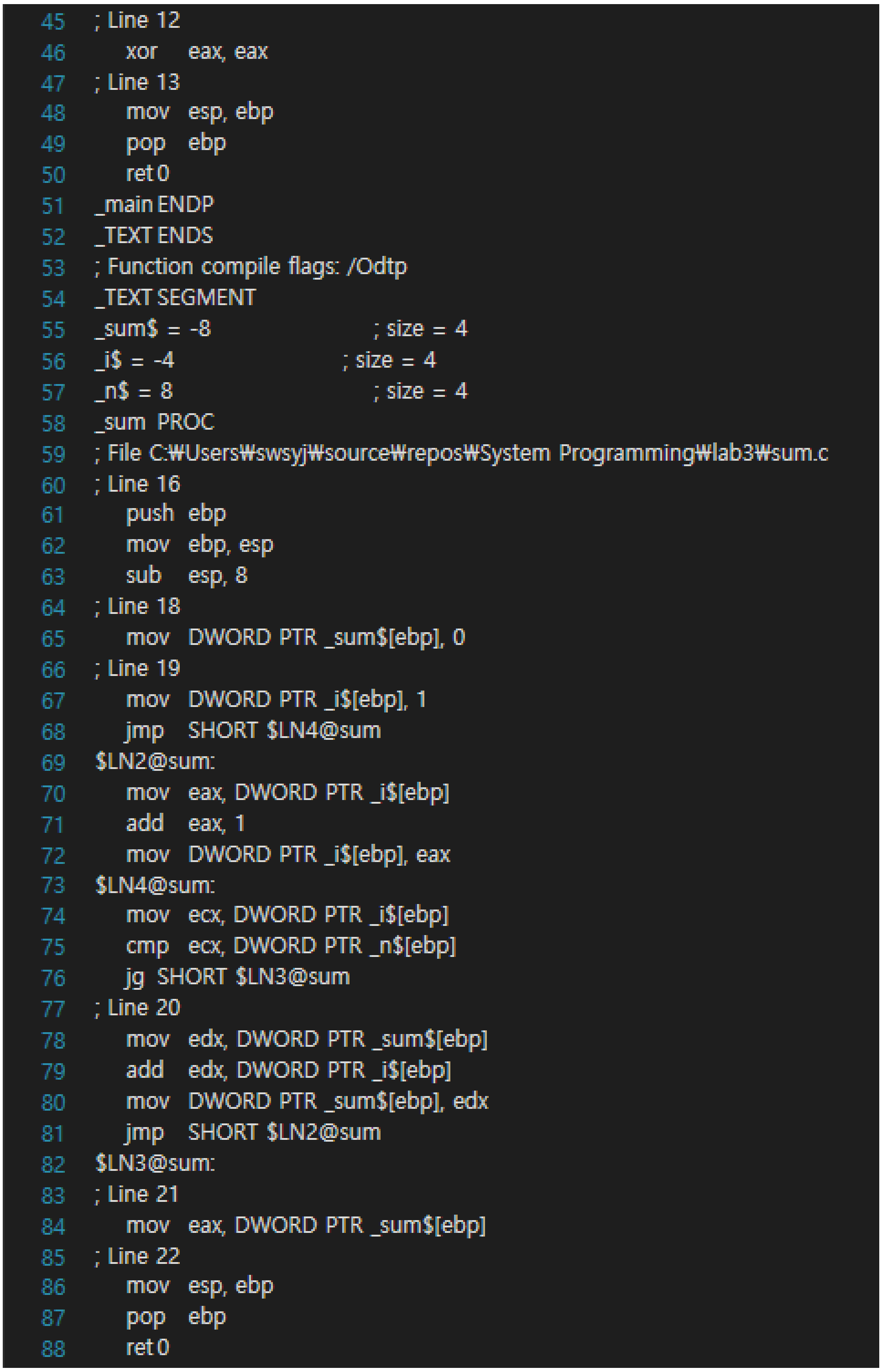


```
C:₩Users₩swsyj₩source₩repos₩System Programming₩lab3>cl /FA sum.c
x86용 Microsoft (R) C/C++ 최적화 컴파일러 버전 19.29.30146
Copyright (c) Microsoft Corporation. All rights reserved.

sum.c
Microsoft (R) Incremental Linker Version 14.29.30146.0
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:sum.exe
sum.obj
```

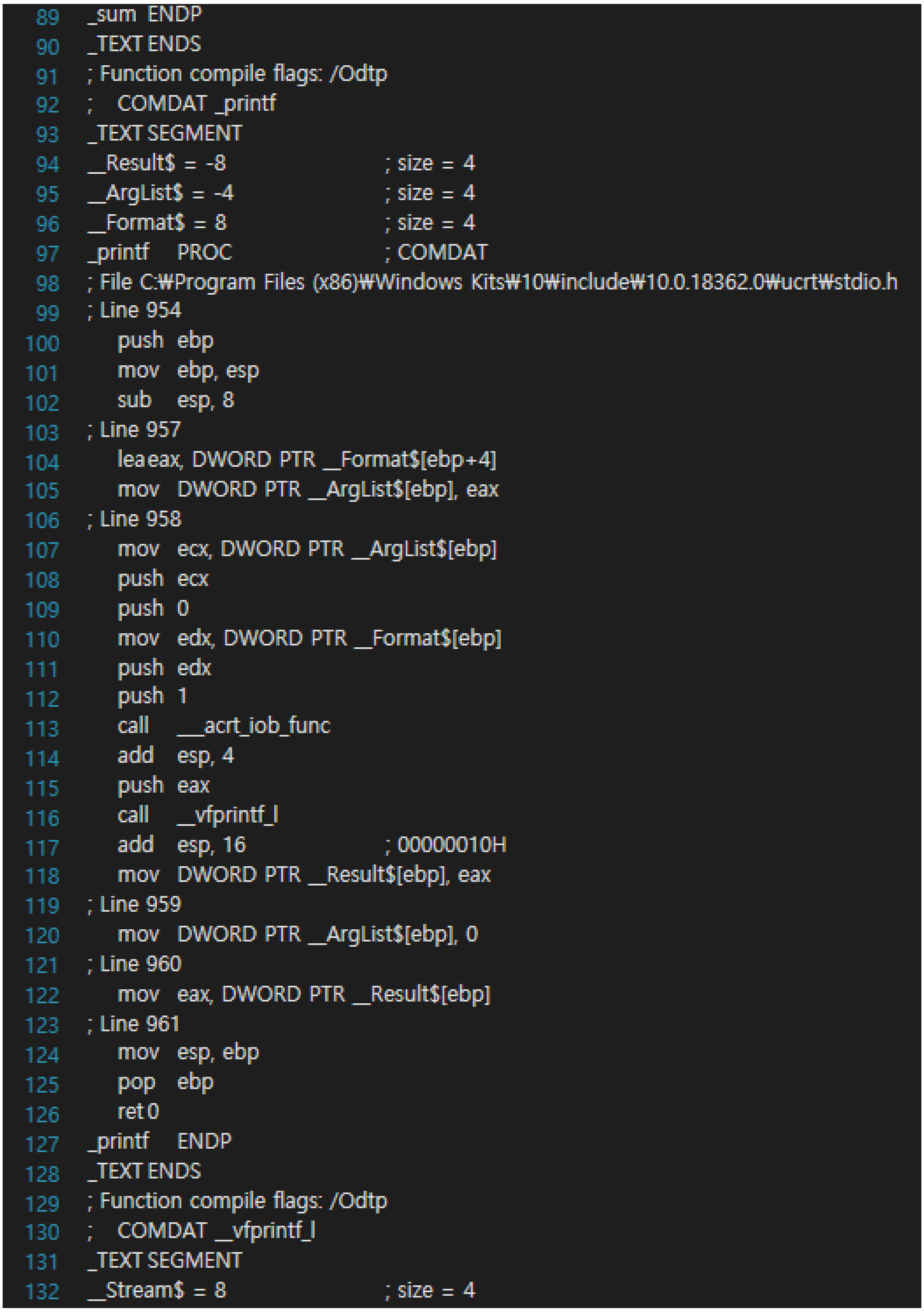
-VS 파일에 sum.c를 만들고 C컴파일러 옵션을 조절하여 sum.asm을 생성함



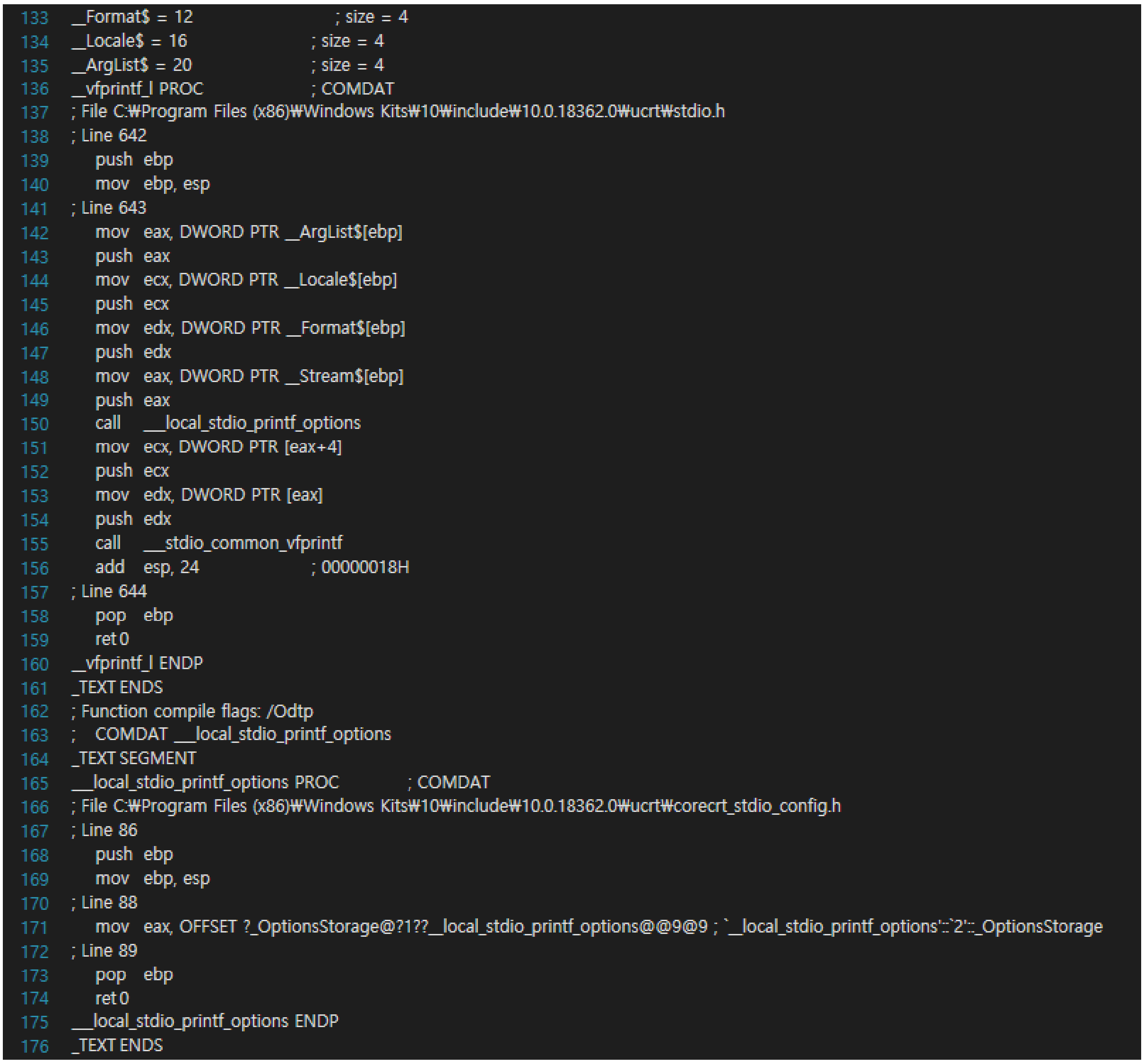
```
; Listing generated by Microsoft (R) Optimizing Compiler Version 19.29.30146.0

	TITLE	C:\Users\swsyj\source\repos\System Programming\lab3\sum.obj
	.686P
	.XMM
	include listing.inc
	.model	flat

INCLUDELIB LIBCMT
INCLUDELIB OLDNAMES

PUBLIC	___local_stdio_printf_options
PUBLIC	__vfprintf_l
PUBLIC	_printf
PUBLIC	_sum
PUBLIC	_main
EXTRN	___acrt_iob_func:PROC
EXTRN	___stdio_common_vfprintf:PROC
_DATA	SEGMENT
COMM	?_OptionsStorage@?1??__local_stdio_printf_options@@9@9:QWORD
_DATA	ENDS
_DATA	SEGMENT
$SG9253	DB	'%d', 0aH, 00H
_DATA	ENDS
; Function compile flags: /Odtp
_TEXT	SEGMENT
_s$ = -4						; size = 4
_main	PROC
; File C:\Users\swsyj\source\repos\System Programming\lab3\sum.c
; Line 6
	push	ebp
	mov	ebp, esp
	push	ecx
; Line 9
	push	10					; 0000000aH
	call	_sum
	add	esp, 4
	mov	DWORD PTR _s$[ebp], eax
; Line 10
	mov	eax, DWORD PTR _s$[ebp]
	push	eax
	push	OFFSET $SG9253
	call	_printf
	add	esp, 8
```



```
; Line 12
    xor    eax, eax
; Line 13
    mov   esp, ebp
    pop   ebp
    ret 0
_main ENDP
_TEXT ENDS
; Function compile flags: /Odtp
_TEXT SEGMENT
_sum$ = -8                      ; size = 4
_i$ = -4                     ; size = 4
_n$ = 8                         ; size = 4
_sum  PROC
; File C:₩Users₩swsyj₩source₩repos₩System Programming₩lab3₩sum.c
; Line 16
    push  ebp
    mov   ebp, esp
    sub    esp, 8
; Line 18
    mov   DWORD PTR _sum$[ebp], 0
; Line 19
    mov   DWORD PTR _i$[ebp], 1
    jmp   SHORT $LN4@sum
$LN2@sum:
    mov   eax, DWORD PTR _i$[ebp]
    add   eax, 1
    mov   DWORD PTR _i$[ebp], eax
$LN4@sum:
    mov   ecx, DWORD PTR _i$[ebp]
    cmp   ecx, DWORD PTR _n$[ebp]
    jg  SHORT $LN3@sum
; Line 20
    mov   edx, DWORD PTR _sum$[ebp]
    add   edx, DWORD PTR _i$[ebp]
    mov   DWORD PTR _sum$[ebp], edx
    jmp   SHORT $LN2@sum
$LN3@sum:
; Line 21
    mov   eax, DWORD PTR _sum$[ebp]
; Line 22
    mov   esp, ebp
    pop   ebp
    ret 0
```



```
_sum	ENDP
_TEXT	ENDS
; Function compile flags: /Odtp
;	COMDAT _printf
_TEXT	SEGMENT
__Result$ = -8					; size = 4
__ArgList$ = -4					; size = 4
__Format$ = 8					; size = 4
_printf	PROC					; COMDAT
; File C:₩Program Files (x86)₩Windows Kits₩10₩include₩10.0.18362.0₩ucrt₩stdio.h
; Line 954
	push	ebp
	mov	ebp, esp
	sub	esp, 8
; Line 957
	lea	eax, DWORD PTR __Format$[ebp+4]
	mov	DWORD PTR __ArgList$[ebp], eax
; Line 958
	mov	ecx, DWORD PTR __ArgList$[ebp]
	push	ecx
	push	0
	mov	edx, DWORD PTR __Format$[ebp]
	push	edx
	push	1
	call	___acrt_iob_func
	add	esp, 4
	push	eax
	call	__vfprintf_l
	add	esp, 16					; 00000010H
	mov	DWORD PTR __Result$[ebp], eax
; Line 959
	mov	DWORD PTR __ArgList$[ebp], 0
; Line 960
	mov	eax, DWORD PTR __Result$[ebp]
; Line 961
	mov	esp, ebp
	pop	ebp
	ret	0
_printf	ENDP
_TEXT	ENDS
; Function compile flags: /Odtp
;	COMDAT __vfprintf_l
_TEXT	SEGMENT
__Stream$ = 8					; size = 4
```



```
__Format$ = 12                          ; size = 4
__Locale$ = 16                      ; size = 4
_ArgList$ = 20                      ; size = 4
__vfprintf_l PROC                   ; COMDAT
; File C:₩Program Files (x86)₩Windows Kits₩10₩include₩10.0.18362.0₩ucrt₩stdio.h
; Line 642
    push  ebp
    mov   ebp, esp
; Line 643
    mov   eax, DWORD PTR _ArgList$[ebp]
    push  eax
    mov   ecx, DWORD PTR __Locale$[ebp]
    push  ecx
    mov   edx, DWORD PTR __Format$[ebp]
    push  edx
    mov   eax, DWORD PTR __Stream$[ebp]
    push  eax
    call  ___local_stdio_printf_options
    mov   ecx, DWORD PTR [eax+4]
    push  ecx
    mov   edx, DWORD PTR [eax]
    push  edx
    call  ___stdio_common_vfprintf
    add   esp, 24                   ; 00000018H
; Line 644
    pop   ebp
    ret 0
__vfprintf_l ENDP
_TEXT ENDS
; Function compile flags: /Odtp
;   COMDAT ___local_stdio_printf_options
_TEXT SEGMENT
___local_stdio_printf_options PROC              ; COMDAT
; File C:₩Program Files (x86)₩Windows Kits₩10₩include₩10.0.18362.0₩ucrt₩corecrt_stdio_config.h
; Line 86
    push  ebp
    mov   ebp, esp
; Line 88
    mov   eax, OFFSET ?_OptionsStorage@?1??__local_stdio_printf_options@@9@9 ; `__local_stdio_printf_options'::`2'::_OptionsStorage
; Line 89
    pop   ebp
    ret 0
___local_stdio_printf_options ENDP
_TEXT ENDS
```

-강의자료 7장 5페이지의 c2asm.asm와 같은 형태로 생성된 걸 볼 수 있다.