;; Author: Marco Martinez

;; Program: IntegerExpressionCalculation.asm

;; Date: 10/7/2018

;; Purpose: To calculate the expression: A = (A+B)-(C+D)

;;

;; Software Change Record

;; Name Date What

;; Marco 10/7 Baseline for integer calculation A = (A+B) - (C-D)

;;

.386

.model flat,stdcall

.stack 4096

ExitProcess proto,dwExitCode:dword

.data

valA SDWORD 10

valB SDWORD 15

valC SDWORD 5

valD SDWORD 10

.stack

.code

main proc

mov eax, valA

mov ebx, valB

add eax, ebx

mov ecx, valC

mov edx, valD

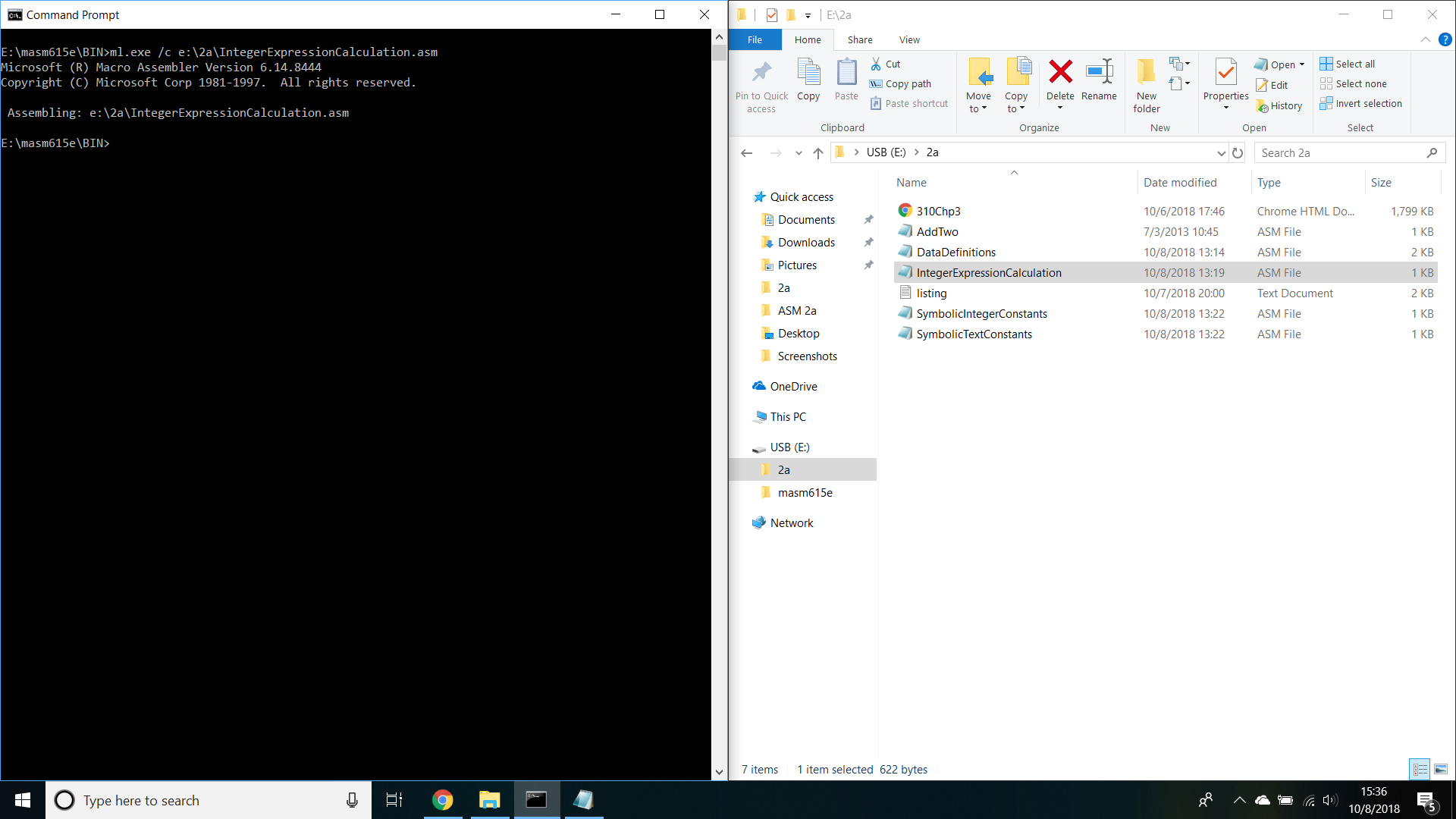
add ecx, edx

sub eax, ecx

invoke ExitProcess,0

main endp

end main



;; Author: Marco Martinez

;; Program: SymbolicIntegerConstants.asm

;; Date: 10/7/2018

;; Purpose: Write a program that defines symbolic constants for all seven days of the week.

;; Create an array variable that uses the symbols as initializers.

;;

;; Software Change Record

;; Name Date What

;; Marco 10/7 Baseline for SymbolicIntegerConstants.asm

;;

.386

.model flat,stdcall

.stack 4096

ExitProcess proto,dwExitCode:dword

MONDAY = 1

TUESDAY = 2

WEDNESDAY = 3

THURSDAY = 4

FRIDAY = 5

SATURDAY = 6

SUNDAY = 7

.data

array BYTE MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

.stack

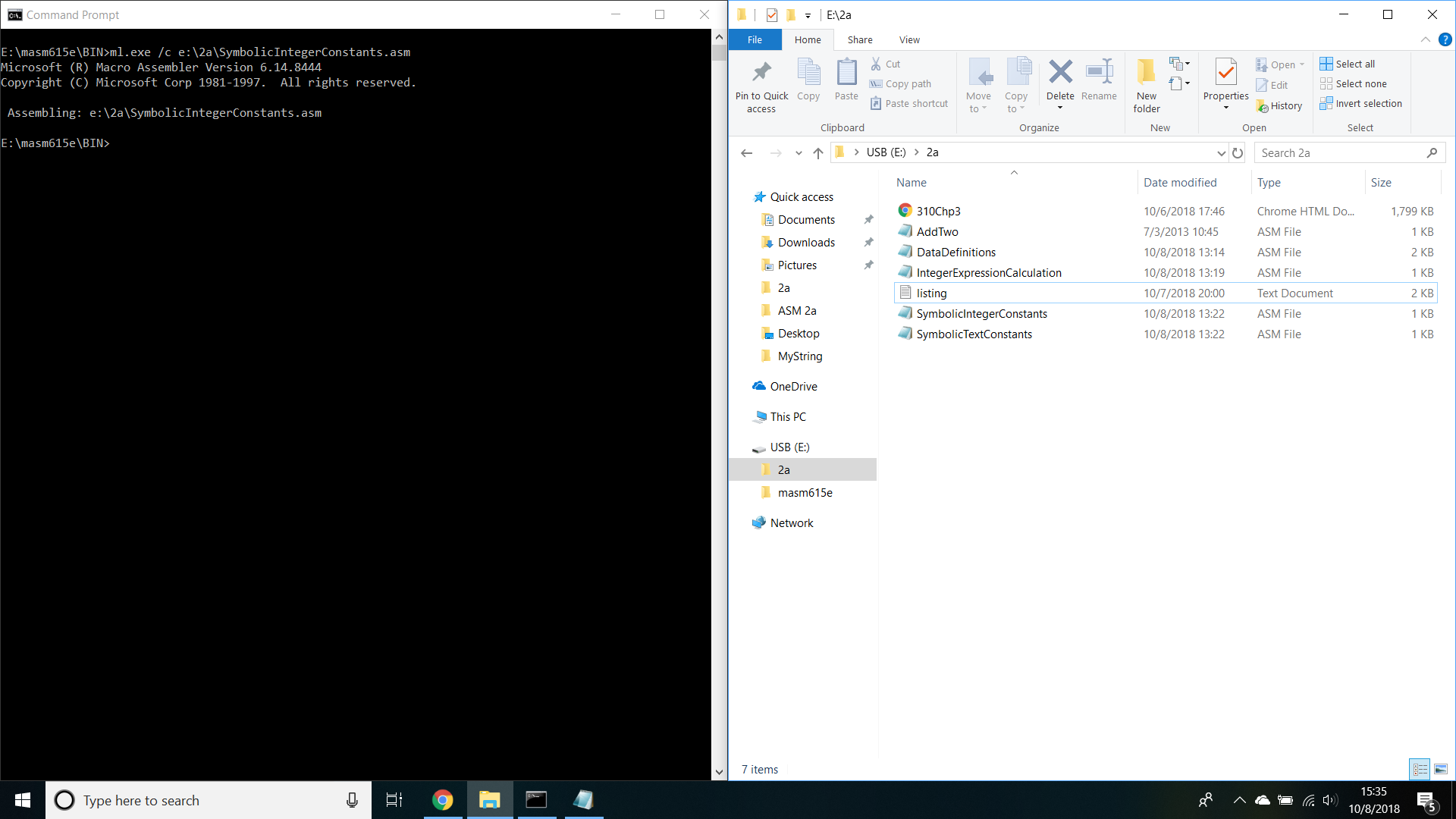
.code

main proc

invoke ExitProcess,0

main endp

end main



;; Author: Marco Martinez

;; Program: DataDefinitions.asm

;; Date: 10/7/2018

;; Purpose: Write a program that contains a definition of each data type listed in Table 3-2 in Section 3.4.

;; Initialize each variable to a value that is consistent with its data type.

;;

;; Software Change Record

;; Name Date What

;; Marco 10/7 Baseline for DataDefinitions.asm

;;

.386

.model flat,stdcall

.stack 4096

ExitProcess proto,dwExitCode:dword

.data

valByte BYTE 255 ; 8bit unsigned integer

valSByte SBYTE -128 ; 8bit signed integer

valWord WORD 65535 ; 16bit unsigned integer

valSWord SWORD -32768 ; 16bit signed integer

valDWord DWORD 4294967295 ; 32bit unsigned integer

valSDWord SDwORD -2147483648 ; 32bit signed integer

valFWord FWORD 1 ; 48bit integer (Far pointer in protected mode)

valQWord QWORD 1 ; 64bit integer

valTByte TBYTE 1 ; 80bit integer

valReal4 REAL4 4.5E4 ; 32bit real, IEEE short real

valReal8 REAL8 5.3E8 ; 64bit real, IEEE long real

valReal10 REAL10 6.7E10 ; 80bit real, IEEE extended real

.stack

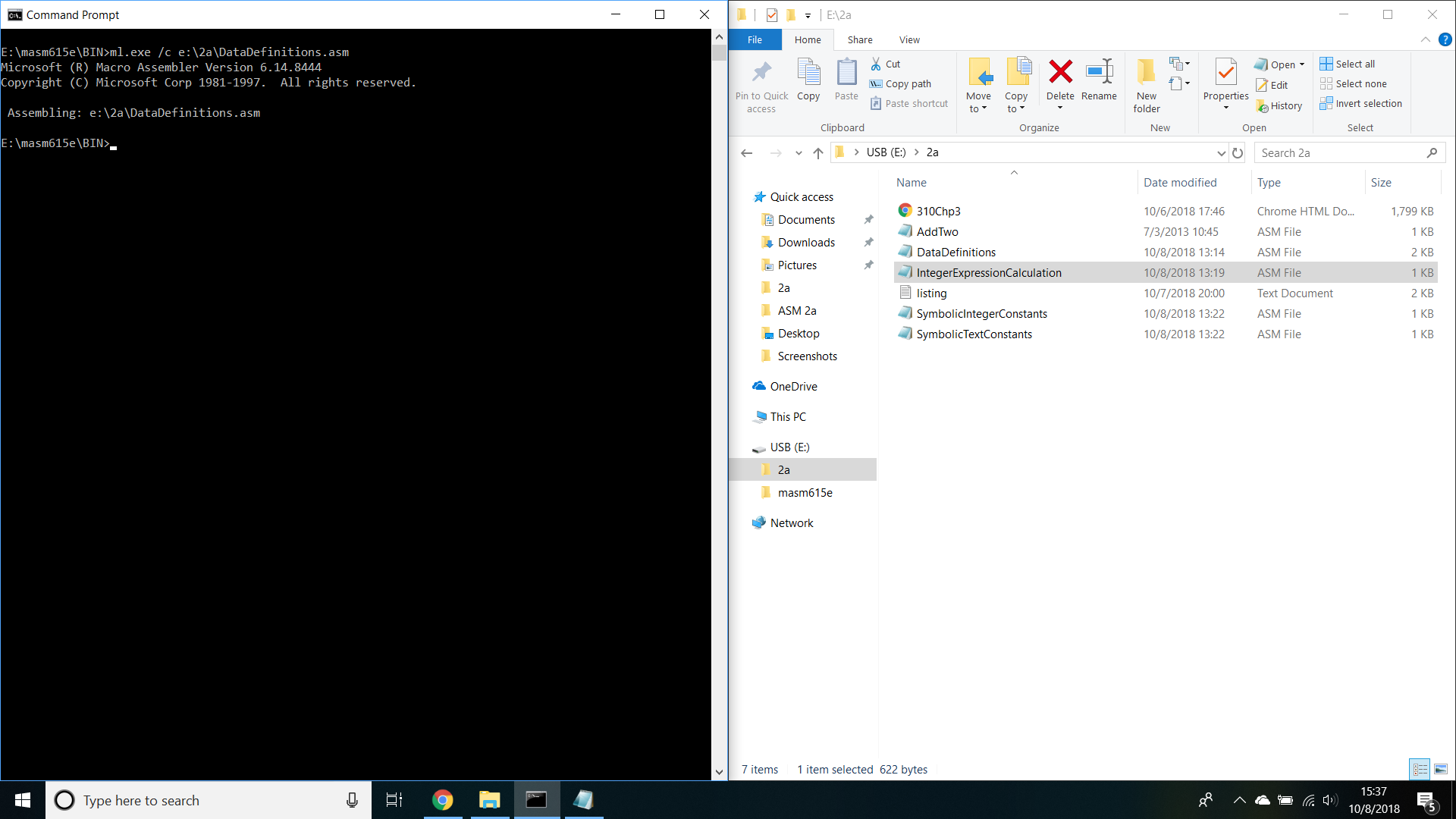
.code

main proc

invoke ExitProcess,0

main endp

end main



;; Author: Marco Martinez

;; Program: SymbolicTextConstants.asm

;; Date: 10/7/2018

;; Purpose: Write a program that defines sumbolic names for several string literals (characters between quotes).

;; Use each symbolic name in a variable defintion.

;;

;; Software Change Record

;; Name Date What

;; Marco 10/7 Baseline for SymbolicTextConstants.asm

;;

.386

.model flat,stdcall

.stack 4096

ExitProcess proto,dwExitCode:dword

MONDAY EQU <'Monday',0>

TUESDAY EQU <'Tuesday',0>

WEDNESDAY EQU <'Wednesday',0>

THURSDAY EQU <'Thursday',0>

FRIDAY EQU <'Friday',0>

SATURDAY EQU <'Saturday',0>

SUNDAY EQU <'Sunday',0>

.data

array BYTE MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

.stack

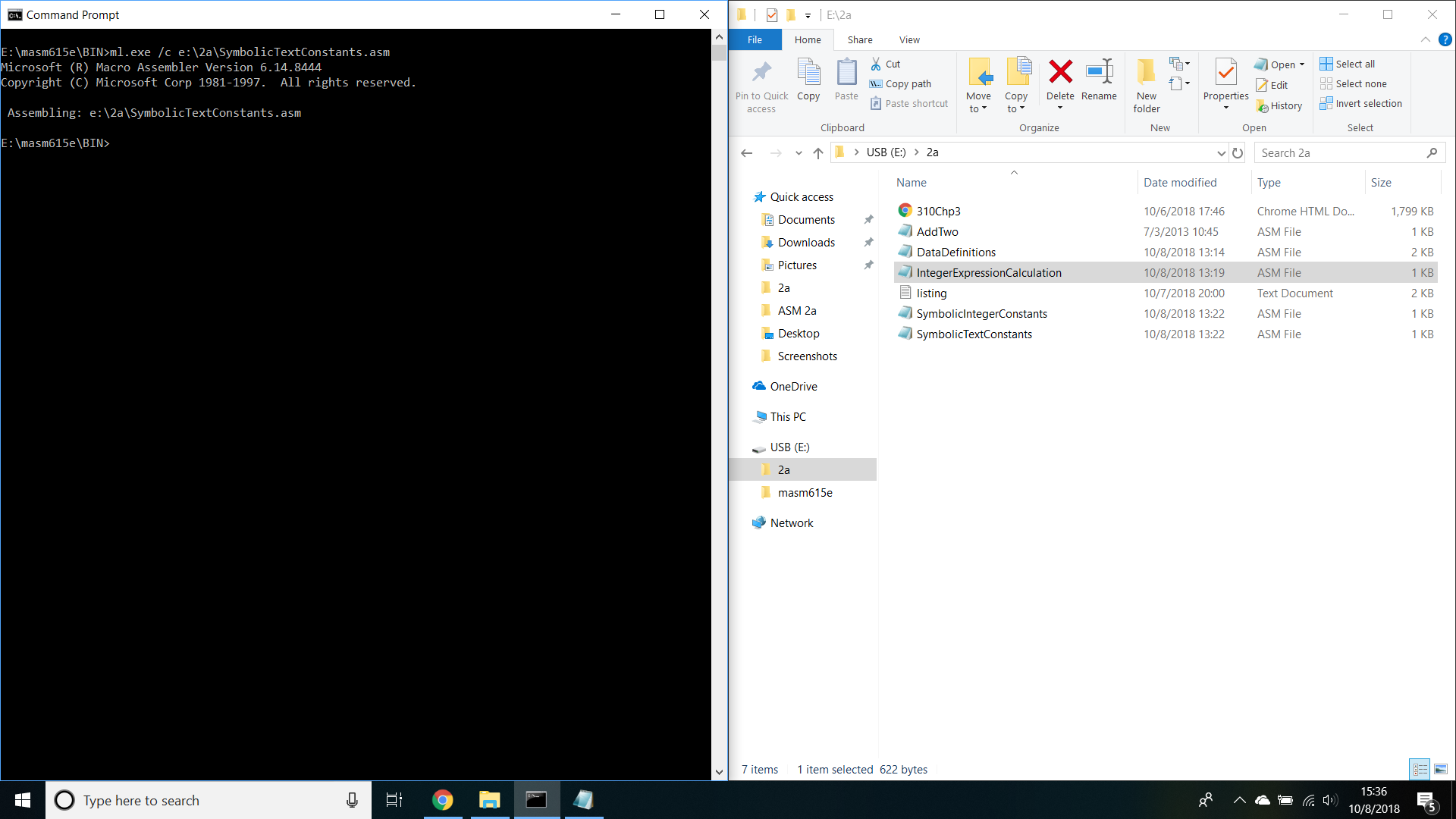
.code

main proc

invoke ExitProcess,0

main endp

end main



Microsoft (R) Macro Assembler Version 6.14.8444 10/07/18 20:00:31

e:\2a\AddTwo.asm Page 1 - 1

; AddTwo.asm - adds two 32-bit integers.

; Chapter 3 example

.386

.model flat,stdcall

.stack 4096

ExitProcess proto,dwExitCode:dword

00000000 .code

**(Starting address for program.)**

00000000 main proc

**(Starting address for program.)**

00000000 B8 00000005 mov eax,5

**(The action MOV starts at 00000000 and B8 is the machine code instruction while 00000005 is the constant 32-bit value.)**

00000005 83 C0 06 add eax,6

**(The action ADD starts at the offset "00000005", 83 is the value of ADD, C0 is the value for the EAX register, and 06 is the value of 6.)**

invoke ExitProcess,0

0000000F main endp

**(This address indicates the end of the program as initiated by "invoke ExitProcess.")**

end main

Microsoft (R) Macro Assembler Version 6.14.8444 10/07/18 20:00:31

e:\2a\AddTwo.asm Symbols 2 - 1

Segments and Groups:

N a m e Size Length Align Combine Class

FLAT . . . . . . . . . . . . . . GROUP

STACK . . . . . . . . . . . . . 32 Bit 00001000 DWord Stack 'STACK'

\_DATA . . . . . . . . . . . . . 32 Bit 00000000 DWord Public 'DATA'

\_TEXT . . . . . . . . . . . . . 32 Bit 0000000F DWord Public 'CODE'

Procedures, parameters and locals:

N a m e Type Value Attr

ExitProcess . . . . . . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL

main . . . . . . . . . . . . . . P Near 00000000 \_TEXT Length= 0000000F Public STDCALL

Symbols:

N a m e Type Value Attr

@CodeSize . . . . . . . . . . . Number 00000000h

@DataSize . . . . . . . . . . . Number 00000000h

@Interface . . . . . . . . . . . Number 00000003h

@Model . . . . . . . . . . . . . Number 00000007h

@code . . . . . . . . . . . . . Text \_TEXT

@data . . . . . . . . . . . . . Text FLAT

@fardata? . . . . . . . . . . . Text FLAT

@fardata . . . . . . . . . . . . Text FLAT

@stack . . . . . . . . . . . . . Text FLAT

0 Warnings

0 Errors

