

1. Adunări, Scăderi BYTE – Ex. 7

$c-(d+d+d)+(a-b)$

a: [401000h] = (B) 4 `MOV AL, BYTE PTR DS:[401002] ; AL = c = 15`
b: [401001h] = (B) 1 `SUB AL, BYTE PTR DS:[401003] ; AL = AL-d = 12`
c: [401002h] = (B) 15 `SUB AL, BYTE PTR DS:[401003] ; AL = AL-d = 9`
d: [401003h] = (B) 3 `SUB AL, BYTE PTR DS:[401003] ; AL = AL-d = 6`
 `MOV BL, BYTE PTR DS:[401001] ; BL = b = 1`
 `SUB BYTE PTR DS:[401000], BL ; a = a-BL = 4-1 = 3`
 `ADD AL, BYTE PTR DS:[401000] ; AL = AL+a = 6+3 = 9`

The screenshot shows a debugger window titled "CPU - main thread, module as_b_7". The assembly window displays the following instructions:

```

00402000  A0 02104000 MOV AL, BYTE PTR DS:[401002]
00402005  2A05 03104000 SUB AL, BYTE PTR DS:[401003]
00402006  2A05 03104000 SUB AL, BYTE PTR DS:[401003]
00402011  2A05 03104000 SUB AL, BYTE PTR DS:[401003]
00402017  8A1D 01104000 MOV BL, BYTE PTR DS:[401001]
0040201D  281D 00104000 SUB BYTE PTR DS:[401000], BL
00402023  0205 00104000 ADD AL, BYTE PTR DS:[401000]
00402029  6A 00        PUSH 0
0040202B  FF15 3C304000 CALL DWORD PTR DS:[<msvcrt.exit>]
00402031  0000        ADD BYTE PTR DS:[EAX], AL
00402033  0000        ADD BYTE PTR DS:[EAX], AL
00402035  0000        ADD BYTE PTR DS:[EAX], AL
00402037  0000        ADD BYTE PTR DS:[EAX], AL
00402039  0000        ADD BYTE PTR DS:[EAX], AL
0040203B  0000        ADD BYTE PTR DS:[EAX], AL

```

The Stack window shows the current stack frame for "as_b_7.<ModuleEntryPoint>+29".

The Registers (FPU) window shows the following values:

Register	Value
EAX	0019FF09
ECX	00402000 as_b_7.<ModuleEntryPoint>
EDX	00402000 as_b_7.<ModuleEntryPoint>
EBX	0034F001
ESP	0019FF74
EBP	0019FF80
ESI	00402000 as_b_7.<ModuleEntryPoint>
EDI	00402000 as_b_7.<ModuleEntryPoint>
EIP	00402029 as_b_7.00402029

The Disassembly window shows the instruction at address 0019FF74: `RETURN to KERNEL32.BaseThreadInitTh`.

2. Adunări, Scăderi BYTE – Ex. 24

$$(a-b-b-c)+(a-c-c-d)$$

a: [401000h] = (B) 7
b: [401001h] = (B) 3
c: [401002h] = (B) 2
d: [401003h] = (B) 4

```

MOV BL, BYTE PTR DS:[401001] ; BL = b = 3
ADD BL, BL                    ; BL = 3+3 = 6
MOV CL, BYTE PTR DS:[401002] ; CL = c = 2
ADD BYTE PTR DS:[401003], CL ; d = d+CL = 4+2 = 6
ADD CL, CL                    ; CL = 2+2 = 4
MOV AL, BYTE PTR DS:[401000] ; AL = a = 7
ADD AL, AL                    ; AL = 7+7 = 14
SUB AL, BL                    ; AL = AL - 6 = 8
SUB AL, CL                    ; AL = AL - 4 = 4
SUB AL, BYTE PTR DS:[401003] ; AL = AL - d = 4-6 = -2 ($FE)

```

The screenshot shows a debugger window titled "CPU - main thread, module as_b_7". The assembly window displays the following instructions:

```

00402000 8A1D 010400 MOV BL, BYTE PTR DS:[401001]
00402006 000B ADD BL, BL
00402008 8A0D 021040 MOV CL, BYTE PTR DS:[401002]
0040200E 000D 031040 ADD BYTE PTR DS:[401003], CL
00402014 00C9 ADD CL, CL
00402016 A0 001040 MOV AL, BYTE PTR DS:[401000]
00402018 00C0 ADD AL, AL
0040201D 28D8 SUB AL, BL
0040201F 28C8 SUB AL, CL
00402021 2A05 031040 SUB AL, BYTE PTR DS:[401003]
00402027 6A 00 PUSH 0
00402029 FF15 3C3040 CALL DWORD PTR DS:[<msvcrt.exit>]
0040202F 0000 ADD BYTE PTR DS:[EAX], AL
00402031 0000 ADD BYTE PTR DS:[EAX], AL
00402033 0000 ADD BYTE PTR DS:[EAX], AL

```

The Registers (FPU) window shows the following values:

Register	Value
EAX	0019FFFE
ECX	00402004 as_b_7.00402004
EDX	00402000 as_b_7.<ModuleEntryP
EBX	0037B006 as_b_7.<ModuleEntryP
ESP	0019FF74
EBP	0019FF80
ESI	00402000 as_b_7.<ModuleEntryP
EDI	00402000 as_b_7.<ModuleEntryP
EIP	00402027 as_b_7.00402027

The Stack window shows the following values:

Address	Hex dump
00401000	07 03 02 06 00 00 00 00 00 00 00 00 00 00 00 00
00401010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004011A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

The Memory dump window shows the following values:

Address	Hex dump
0019FF74	76BC6359
0019FF78	0037B000
0019FF7C	76BC6340
0019FF80	0019FFDC
0019FF84	77BD7C14
0019FF88	0037B000
0019FF8C	7CC89ED5
0019FF90	00000000
0019FF94	00000000
0019FF98	0037B000
0019FF9C	00000000
0019FFA0	00000000
0019FFA4	00000000
0019FFA8	00000000
0019FFAC	00000000
0019FFB0	00000000
0019FFB4	00000000
0019FFB8	00000000
0019FFBC	00000000
0019FFC0	00000000
0019FFC4	0019FF8C
0019FFC8	00000000
0019FFCC	0019FFE4
0019FFD0	77BEA040
0019FFD4	0B1602A1
0019FFD8	00000000
0019FFDC	0019FFEC
0019FFE0	77BD7BE4
0019FFE4	55555555

3. Adunări, Scăderi WORD – Ex. 7

(c+c+c)-b+(d-a)

a: [401000h] = (W)500 MOV AX, WORD PTR DS:[401004] ; AX = c = 300
b: [401002h] = (W)200 ADD AX, WORD PTR DS:[401004] ; AX = AX + c = 600
c: [401004h] = (W)300 ADD AX, WORD PTR DS:[401004] ; AX = AX + c = 900
d: [401006h] = (W)400 SUB AX, WORD PTR DS:[401002] ; AX = AX - b = 900 - 200 = 700
 ADD AX, WORD PTR DS:[401006] ; AX = AX + d = 700 + 400 = 1100
 SUB AX, WORD PTR DS:[401000] ; AX = AX - a = 1100 - 500 = 600

CPU - main thread, module as_b_7

Address	Hex dump	Assembly
00402000	66:A1 041040	MOV AX, WORD PTR DS:[401004]
00402006	66:0305 0410	ADD AX, WORD PTR DS:[401004]
0040200D	66:0305 0410	ADD AX, WORD PTR DS:[401004]
00402014	66:2B05 0210	SUB AX, WORD PTR DS:[401002]
0040201B	66:0305 0610	ADD AX, WORD PTR DS:[401006]
00402022	66:2B05 0010	SUB AX, WORD PTR DS:[401000]
00402029	6A 00	PUSH 0
0040202B	FF15 3C304000	CALL DWORD PTR DS:[<msvcrt.exit>]
00402031	0000	ADD BYTE PTR DS:[EAX], AL
00402033	0000	ADD BYTE PTR DS:[EAX], AL
00402035	0000	ADD BYTE PTR DS:[EAX], AL
00402037	0000	ADD BYTE PTR DS:[EAX], AL
00402039	0000	ADD BYTE PTR DS:[EAX], AL
0040203B	0000	ADD BYTE PTR DS:[EAX], AL
0040203D	0000	ADD BYTE PTR DS:[EAX], AL

Stack [0019FF70]=0
Imm=0

as_b_7.<ModuleEntryPoint>+29

Address	Hex dump	Comment
00401000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004010A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004010B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004010C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004010D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004010E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004010F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00401190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
004011A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Registers (FPU)

Register	Value	Comment
EAX	00190258	
ECX	00402000	as_b_7.<ModuleEntryPoint>
EDX	00402000	as_b_7.<ModuleEntryPoint>
EBX	0031D000	
ESP	0019FF74	
EBP	0019FF80	
ESI	00402000	as_b_7.<ModuleEntryPoint>
EDI	00402000	as_b_7.<ModuleEntryPoint>
EIP	00402029	as_b_7.00402029
C 0	ES 002B	32bit 0(FFFFFFFF)
P 0	CS 0023	32bit 0(FFFFFFFF)
A 0	SS 002B	32bit 0(FFFFFFFF)
Z 0	DS 002B	32bit 0(FFFFFFFF)
S 0	FS 0053	32bit 320000(FEE)

Modify EAX

	EAX	AX	Al
Hexadecimal	00190258	0258	02
Signed	1639000	600	2
Unsigned	1639000	600	2
Character			\x02

OK

0019FFB8 00000000
0019FFBC 00000000
0019FFC0 00000000
0019FFC4 0019FF8C
0019FFC8 00000000
0019FFCC 0019FFE4
0019FFD0 77BEA040
0019FFD4 E6604ED2
0019FFD8 00000000
0019FFDC 0019FFEC
0019FFE0 77BD7BE4
0019FFE4 FFFFFFFF

Pointer to next SEH record
SE handler
RETURN from ntdll.77BD7BE4
End of SEH chain

4. Adunări, Scăderi WORD – Ex. 24

(a-c)+(b-d)

a: [401000h] = (W)2048 `MOV AX, WORD PTR DS:[401000]` ; AX = a = 800h
b: [401002h] = (W) 512 `SUB AX, WORD PTR DS:[401004]` ; AX = AX - c = 800h-400h=400h
c: [401004h] = (W)1024 `ADD AX, WORD PTR DS:[401002]` ; AX = AX + b = 400h+200h=600h
d: [401006h] = (W) 256 `SUB AX, WORD PTR DS:[401006]` ; AX = AX - d = 600h-100h=500h

CPU - main thread, module as_b_7

Address	Hex dump	Assembly
00402000	66:A1 001040	<code>MOV AX, WORD PTR DS:[401000]</code>
00402006	66:2B05 0410	<code>SUB AX, WORD PTR DS:[401004]</code>
0040200D	66:0305 0210	<code>ADD AX, WORD PTR DS:[401002]</code>
00402014	66:2B05 0610	<code>SUB AX, WORD PTR DS:[401006]</code>
0040201B	6A 00	<code>PUSH 0</code>
0040201D	FF15 3C304000	<code>CALL DWORD PTR DS:[&msvcrt.exit]</code>
00402023	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
00402025	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
00402027	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
00402029	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
0040202B	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
0040202D	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
0040202F	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
00402031	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>
00402033	0000	<code>ADD BYTE PTR DS:[EAX], AL</code>

Stack [0019FF70]=0
Imm=0

as_b_7.<ModuleEntryPoint>+1B

Address	Hex dump	Registers (FPU)
0019FF74	76BC6359	EAX 00190500
0019FF78	00219000	ECX 00402000 as_b_7.<ModuleEntryPoint>
0019FF7C	76BC6340	EDX 00402000 as_b_7.<ModuleEntryPoint>
0019FF80	0019FFDC	EBX 00219000
0019FF84	77BD7C14	ESP 0019FF74
0019FF88	00219000	EBP 0019FF80
0019FF8C	04FBE49F	ESI 00402000 as_b_7.<ModuleEntryPoint>
0019FF90	00000000	EDI 00402000 as_b_7.<ModuleEntryPoint>
0019FF94	00000000	EIP 0040201B as_b_7.0040201B
0019FF98	00219000	C 0 ES 002B 32bit 0(FFFFFFFF)
0019FF9C	00000000	P 1 CS 0023 32bit 0(FFFFFFFF)
0019FFA0	00000000	A 0 SS 002B 32bit 0(FFFFFFFF)
0019FFA4	00000000	Z 0 DS 002B 32bit 0(FFFFFFFF)
0019FFA8	00000000	S 0 FS 0053 32bit 21C000(FFF)
0019FFAC	00000000	T 0 GS 002B 32bit 0(FFFFFFFF)
0019FFB0	00000000	D 0
0019FFB4	00000000	O 0 LastErr 00000000 ERROR_SUCCE
0019FFB8	00000000	EFL 00000206 (NO NB NE O NS PF GE
0019FFBC	00000000	
0019FFC0	00000000	
0019FFC4	0019FF8C	
0019FFC8	00000000	
0019FFCC	0019FFE4	
0019FFD0	77BEA040	
0019FFD4	A32578EB	
0019FFD8	00000000	
0019FFDC	0019FFEC	
0019FFE0	77BD7BE4	
0019FFE4	55555555	

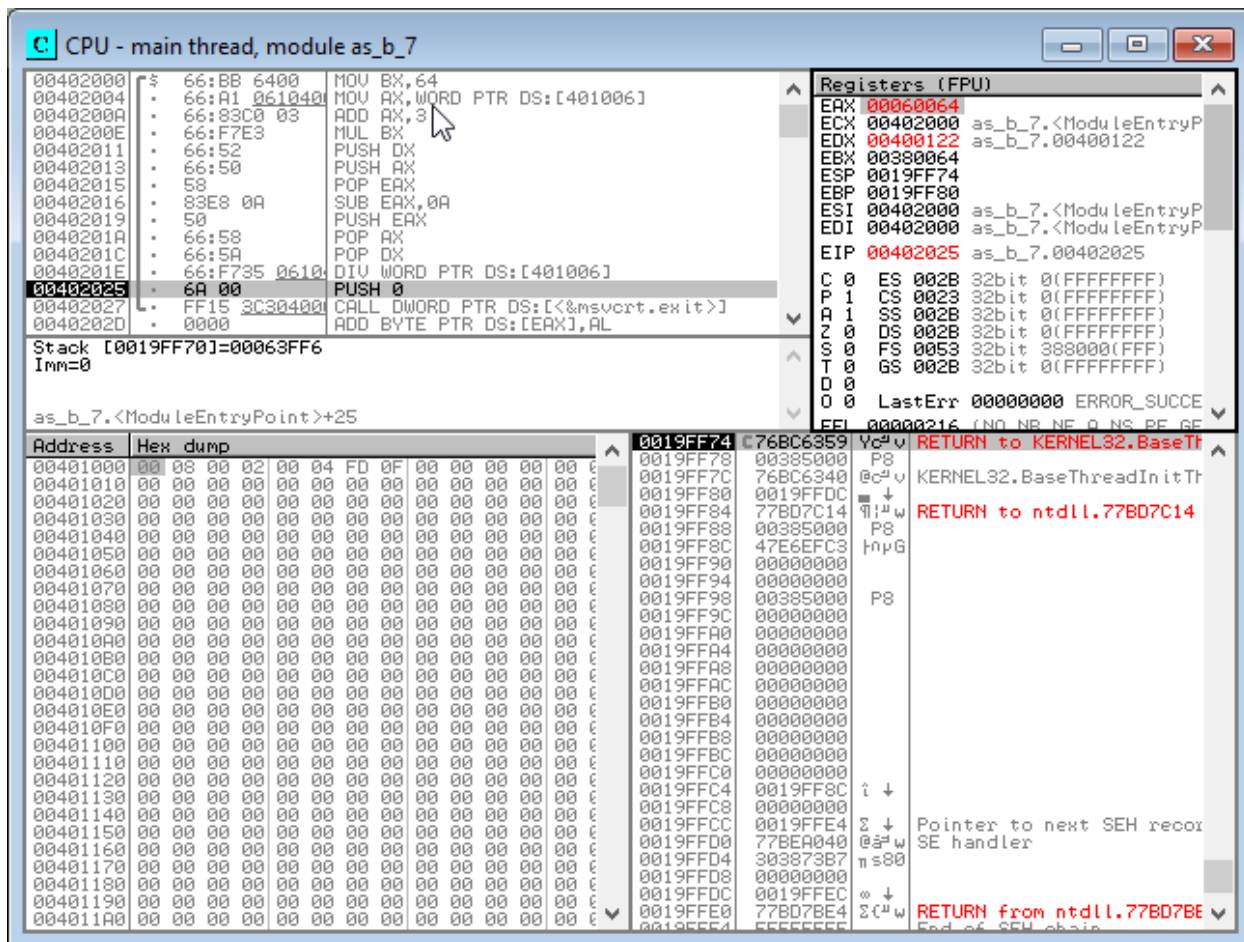
RETURN to KERNEL32.BaseThreadInitThunk
RETURN to ntdll.77BD7C14
RETURN from ntdll.77BD7BE4

5. Înmulțiri, Împărțiri a,b,c BYTE, d WORD – Ex. 7

$[100 \cdot (d+3) - 10] / d$

```

a: [-----] =          MOV  BX, 64          ;
b: [-----] =          MOV  AX, WORD PTR DS:[401006] ; AX = 0FFDh
c: [-----] =          ADD  AX, 3          ; AX = 1000h
d: [401006h] = (w) 4093  MUL  BX          ; DX:AX = 6:4000h
                          PUSH DX          ;
                          PUSH AX          ;
                          POP  EAX         ; EAX = 00064000h
                          SUB  EAX, 0A     ; EAX = 00063FF6h
                          PUSH EAX         ; >> DX:AX = EAX
                          POP  AX         ; AX = 3FF6h
                          POP  DX         ; DX = 0006h
                          DIV  WORD PTR DS:[401006] ; EAX = q = 100, EDX = r = 290
  
```



6. Înmulțiri, Împărțiri a,b,c BYTE, d WORD – Ex. 24

$$(10*a-5*b)+(d-5*c)$$

a: [401000h] = (B) 25 `MOV DX, WORD PTR DS:[401003]` ; DX = d = 500
b: [401001h] = (B) 3 `MOV AL, 0A` ; AL = 10
c: [401002h] = (B) 9 `MUL BYTE PTR DS:[401000]` ; AX = AL * a = 250
d: [401003h] = (W) 500 `ADD DX, AX` ; DX = 750
 `MOV AL, 5` ; AL = 5
 `MUL BYTE PTR DS:[401001]` ; AX = AL * b = 15
 `SUB DX, AX` ; DX = DX - AX = 735
 `MOV AL, 5` ; AL = 5
 `MUL BYTE PTR DS:[401002]` ; AX = AL * c = 45
 `SUB DX, AX` ; DX = DX - AX = 690

CPU - main thread, module as_b_7

00402000	66:8B15 0310	MOV DX,WORD PTR DS:[401003]
00402007	B0 0A	MOV AL,0A
00402009	F625 0010400	MUL BYTE PTR DS:[401000]
0040200F	66:01C2	ADD DX,AX
00402012	B0 05	MOV AL,5
00402014	F625 0110400	MUL BYTE PTR DS:[401001]
0040201A	66:29C2	SUB DX,AX
0040201D	B0 05	MOV AL,5
0040201F	F625 0210400	MUL BYTE PTR DS:[401002]
00402025	66:29C2	SUB DX,AX
00402028	6A 00	PUSH 0
0040202A	FF15 3C30400	CALL DWORD PTR DS:[<msvcrt.exit>]
00402030	0000	ADD BYTE PTR DS:[EAX],AL
00402032	0000	ADD BYTE PTR DS:[EAX],AL
00402034	0000	ADD BYTE PTR DS:[EAX],AL

Stack [0019FF70]=0
Imm=0

as_b_7.<ModuleEntryPoint>+28

Address	Hex dump
00401000	19 03 09 F4 01 00 00 00 00 00 00 00 00 00 00 00
00401010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004011A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Registers (FPU)

EAX	00190020
ECX	00402000 as_b_7.<ModuleEntryPoint>
EDX	004002B2 as_b_7.004002B2
EBX	00254000
ESP	0019FF74
EBP	0019FF80
ESI	00402000 as_b_7.<ModuleEntryPoint>
EDI	00402000 as_b_7.<ModuleEntryPoint>
EIP	00402028 as_b_7.00402028
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)
Z 0	DS 002B 32bit 0(FFFFFFFF)
S 0	FS 0053 32bit 257000(FFF)
T 0	GS 002B 32bit 0(FFFFFFFF)
D 0	
0 0	LastErr 00000000 ERROR_SUCC
EFL	00000206 (NO NB NE O NS PE GE

Modify EDX

	EDX	DX
Hexadecimal	004002B2	02B2
Signed	4194994	690
Unsigned	4194994	690
Character		

OK

0019FFC8	00000000	Σ ↓	Pointer to next SEH recor
0019FFCC	0019FFE4	@3w	SE handler
0019FFD0	77BEA040	1/79	
0019FFD4	39D82FE7		
0019FFD8	00000000		
0019FFDC	0019FFEC	Σ ↓	
0019FFE0	77BD7BE4	Σ ↓	RETURN from ntdll.77BD7BE
0019FFE4	FFFFFFFF		End of SEH chain

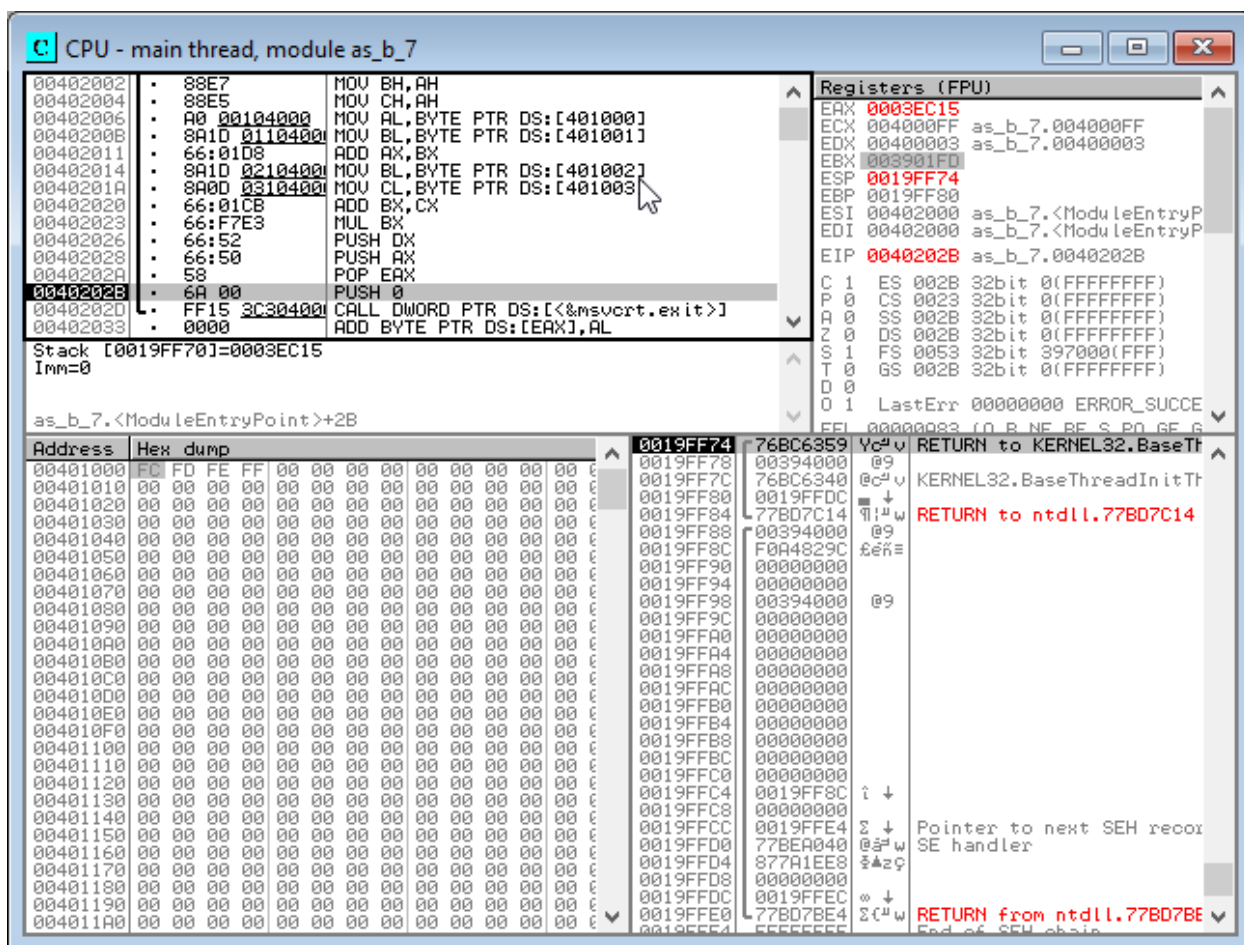
7. Înmulțiri, Împărțiri a,b,c,d BYTE, e,f,g,h WORD – Ex. 7

$$(a+b)*(c+d)$$

```

a: [401000h] = (B) 252      MOV AH, 0           ; AH = BH = CH = 0
b: [401001h] = (B) 253      MOV BH, AH           ;
c: [401002h] = (B) 254      MOV CH, AH           ;
d: [401003h] = (B) 255      MOV AL, BYTE PTR DS:[401000] ; AL = a = 252
                             MOV BL, BYTE PTR DS:[401001] ; BL = b = 253
                             ADD AX, BX             ; AX = a + b = 505
                             MOV BL, BYTE PTR DS:[401002] ; BL = c = 254
                             MOV CL, BYTE PTR DS:[401003] ; CL = d = 255
                             ADD BX, CX             ; BX = c + d = 509
                             MUL BX                 ; DX:AX = 3:EC15h = 257045=505*509
                             PUSH DX                 ;
                             PUSH AX                 ;
                             POP EAX                 ; EAX = DX:AX = 257045

```



8. Înmulțiri, Împărțiri a,b,c,d BYTE, e,f,g,h WORD – Ex. 24

$$[(a-d)+b]*2/c$$

a: [401000h] = (B) 31 `MOV AL, BYTE PTR DS:[401000]` ; AL = a = 31
b: [401001h] = (B) 8 `ADD AL, BYTE PTR DS:[401001]` ; AL = AL + b = 39
c: [401002h] = (B) 4 `SUB AL, BYTE PTR DS:[401003]` ; AL = AL - d = 28
d: [401003h] = (B) 11 `MOV BL, 2` ;
 `MUL BL` ; AX = 28 * 2 = 56
 `DIV BYTE PTR DS:[401002]` ; AX/=c, AL = q = 14, AH = r = 0

