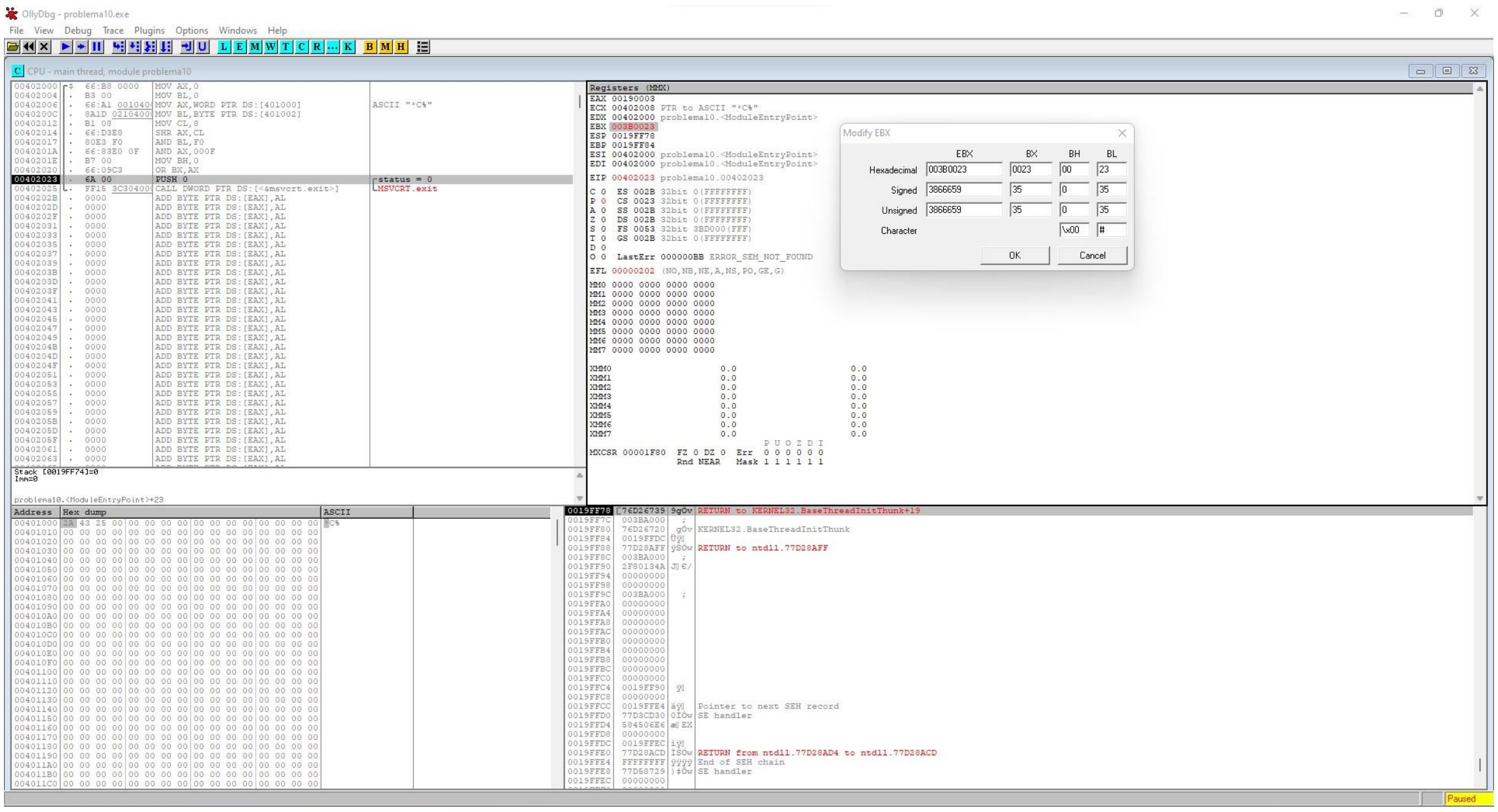
Tema Laborator 4

Suciu Radu – Grupa 216/2

Problema 10

Sa se inlocuiasca bitii 0-3 ai octetului B cu bitii 8-11 ai cuvantului A.

```
bits 32


global start

extern exit


import exit msvcrt.dll

segment data use32 class=data


        a dw 432Ah ; 0100 0011 0010 1010b

        b db 25h   ;           0010 0101b

segment code use32 class=code

    start:


        MOV AX, 0

        MOV BL, 0

        MOV AX, [a]      ; AX = 0100 0011 0010 1010b = 432Ah

        MOV BL, [b]      ; BL =           0010 0101b =   25h


        MOV CL, 8        ; CL = 08h

        SHR AX, CL       ; AX = 0000 0000 0100 0011b = 0043h


        AND BL, 11110000b; BL =      0010 0000b = 20h

        AND AX, 000Fh    ; AX = 0000 0000 0000 0011b = 0003h


        MOV BH, 0        ; BH = 0000 0000b

        OR BX, AX        ; BX = 0000 0000 0010 0011b = 23h

        push    dword 0

        call    [exit]
```


OllyDbg - problema10.exe
File View Debug Trace Plugins Options Windows Help
CPU - main thread, module problema10
Paused
OK
Cancel

```
00402000 ┌$  66:B8 0000     MOV AX,0
00402004 │.  B3 00          MOV BL,0
00402006 │.  66:A1 0010400  MOV AX,WORD PTR DS:[401000]      ASCII "*C%"
0040200C │.  8A1D 0210400   MOV BL,BYTE PTR DS:[401002]
00402012 │.  B1 08          MOV CL,8
00402014 │.  66:D3E8        SHR AX,CL
00402017 │.  80E3 F0        AND BL,F0
0040201A │.  66:83E0 0F     AND AX,000F
0040201E │.  B7 00          MOV BH,0
00402020 │.  66:09C3        OR BX,AX
00402023 │.  6A 00          PUSH 0                            ┌status = 0
00402025 └.  FF15 3C30400   CALL DWORD PTR DS:[<&msvcrt.exit>] └MSVCRT.exit
0040202B  .  0000           ADD BYTE PTR DS:[EAX],AL
0040202D  .  0000           ADD BYTE PTR DS:[EAX],AL
0040202F  .  0000           ADD BYTE PTR DS:[EAX],AL
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
0040203F  .  0000           ADD BYTE PTR DS:[EAX],AL
00402041  .  0000           ADD BYTE PTR DS:[EAX],AL
00402043  .  0000           ADD BYTE PTR DS:[EAX],AL
00402045  .  0000           ADD BYTE PTR DS:[EAX],AL
00402047  .  0000           ADD BYTE PTR DS:[EAX],AL
00402049  .  0000           ADD BYTE PTR DS:[EAX],AL
0040204B  .  0000           ADD BYTE PTR DS:[EAX],AL
0040204D  .  0000           ADD BYTE PTR DS:[EAX],AL
0040204F  .  0000           ADD BYTE PTR DS:[EAX],AL
00402051  .  0000           ADD BYTE PTR DS:[EAX],AL
00402053  .  0000           ADD BYTE PTR DS:[EAX],AL
00402055  .  0000           ADD BYTE PTR DS:[EAX],AL
00402057  .  0000           ADD BYTE PTR DS:[EAX],AL
00402059  .  0000           ADD BYTE PTR DS:[EAX],AL
0040205B  .  0000           ADD BYTE PTR DS:[EAX],AL
0040205D  .  0000           ADD BYTE PTR DS:[EAX],AL
0040205F  .  0000           ADD BYTE PTR DS:[EAX],AL
00402061  .  0000           ADD BYTE PTR DS:[EAX],AL
00402063  .  0000           ADD BYTE PTR DS:[EAX],AL

Stack [0019FF74]=0
Imm=0

problema10.<ModuleEntryPoint>+23
```

```
Registers (MMX)
EAX 00190003
ECX 00402008 PTR to ASCII "*C%"
EDX 00402000 problema10.<ModuleEntryPoint>
EBX 003B0023
ESP 0019FF78
EBP 0019FF84
ESI 00402000 problema10.<ModuleEntryPoint>
EDI 00402000 problema10.<ModuleEntryPoint>

EIP 00402023 problema10.00402023

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 3BD000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr 000000BB ERROR_SEM_NOT_FOUND

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

MM0 0000 0000 0000 0000
MM1 0000 0000 0000 0000
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000

XMM0              0.0                 0.0
XMM1              0.0                 0.0
XMM2              0.0                 0.0
XMM3              0.0                 0.0
XMM4              0.0                 0.0
XMM5              0.0                 0.0
XMM6              0.0                 0.0
XMM7              0.0                 0.0
                                  P U O Z D I
MXCSR 00001F80  FZ 0 DZ 0  Err   0 0 0 0 0 0
                Rnd NEAR   Mask  1 1 1 1 1 1
```

Modify EBX

	EBX	BX	BH	BL
Hexadecimal	003B0023	0023	00	23
Signed	3866659	35	0	35
Unsigned	3866659	35	0	35
Character			\x00	#

```
Address  Hex dump                                         ASCII
00401000 2A 43 25 00 00 00 00 00 00 00 00 00 00 00 00 00 *C%
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
004011B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004011C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

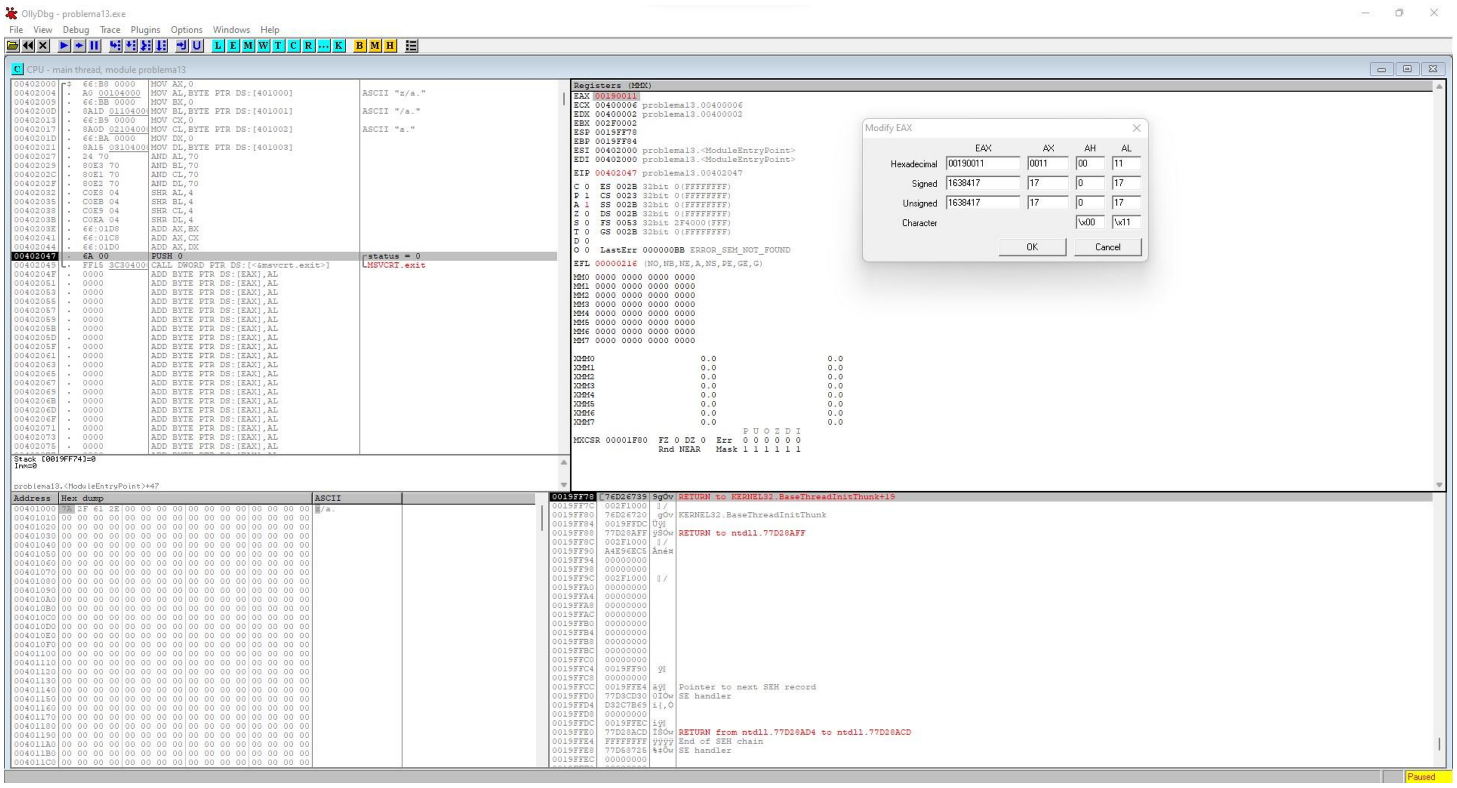
```
0019FF78 ┌76D26739 9gÒv RETURN to KERNEL32.BaseThreadInitThunk+19
0019FF7C  003BA000   ;
0019FF80  76D26720  gÒv KERNEL32.BaseThreadInitThunk
0019FF84  0019FFDC Üÿ
0019FF88  77D28AFF ÿŠÒw RETURN to ntdll.77D28AFF
0019FF8C  003BA000   ;
0019FF90  2F80134A J €/
0019FF94  00000000
0019FF98  00000000
0019FF9C  003BA000   ;
0019FFA0  00000000
0019FFA4  00000000
0019FFA8  00000000
0019FFAC  00000000
0019FFB0  00000000
0019FFB4  00000000
0019FFB8  00000000
0019FFBC  00000000
0019FFC0  00000000
0019FFC4  0019FF90  ÿ
0019FFC8  00000000
0019FFCC  0019FFE4 äÿ   Pointer to next SEH record
0019FFD0  77D3CD30 0ÍÓw SE handler
0019FFD4  584506E6 æ EX
0019FFD8  00000000
0019FFDC  0019FFEC ìÿ
0019FFE0  77D28ACD ÍŠÒw RETURN from ntdll.77D28AD4 to ntdll.77D28ACD
0019FFE4  FFFFFFFF ÿÿÿÿ End of SEH chain
0019FFE8  77D58729 )‡Õw SE handler
0019FFEC  00000000
```

Problema 13

;Dandu-se 4 octeti, sa se obtina in AX suma numerelor intregi ;reprezentate de bitii 4-6 ai celor 4 octeti.

```
bits 32 ; assembling for the 32 bits architecture


; declare the EntryPoint (a label defining the very first instruction of the program)

global start


; declare external functions needed by our program

extern exit               ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll    ; exit is a function that ends the calling process. It is defined in msvcrt.dll

                          ; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions


; our data is declared here (the variables needed by our program)

segment data use32 class=data

        a db 7Ah ; 0111 1010b

        b db 2Fh ; 0010 1111b

        c db 61h ; 0110 0001b

        d db 2Eh ; 0010 1110b

        ;Rezultat final: 17 = 11h = 0001 0001b


segment code use32 class=code
```

```
start:

    MOV AX,0 ; AX = 0000h
    MOV AL, [a] ; AL = 7Ah = 0111 1010b
    MOV BX,0 ; BX = 0000h
    MOV BL, [b] ; BL = 2Fh = 0010 1111b
    MOV CX,0 ; CX = 0000h
    MOV CL, [c] ; CL = 61h = 1110 0001b
    MOV DX,0 ; DX = 0000h
    MOV DL, [d] ; DL = 2Eh = 0010 1110b


    AND AL, 70h ; AL = 0111 0000b = 70h
    AND BL, 70h ; BL = 0010 0000b = 20h
    AND CL, 70h ; CL = 0110 0000b = 60h
    AND DL, 70h ; DL = 0010 0000b = 20h


    SHR AL, 4 ; AL = 0000 0111b = 07h
    SHR BL, 4 ; BL = 0000 0010b = 02h
    SHR CL, 4 ; CL = 0000 0110b = 06h
    SHR DL, 4 ; DL = 0000 0010b = 02h


    ADD AX, BX; AX = AX + BX = 09h
    ADD AX, CX; AX = AX + CX = 0Fh
    ADD AX, DX; AX = AX + DX = 11h


    ; exit(0)
    push    dword 0      ; push the parameter for exit onto the stack
    call    [exit]       ; call exit to terminate the program
```


OllyDbg - problema13.exe

CPU - main thread, module problema13

```
00402000 ┌$  66:B8 0000      MOV AX,0
00402004 │.  A0 00104000     MOV AL,BYTE PTR DS:[401000]          ASCII "z/a."
00402009 │.  66:BB 0000      MOV BX,0
0040200D │.  8A1D 01104000   MOV BL,BYTE PTR DS:[401001]          ASCII "/a."
00402013 │.  66:B9 0000      MOV CX,0
00402017 │.  8A0D 02104000   MOV CL,BYTE PTR DS:[401002]          ASCII "a."
0040201D │.  66:BA 0000      MOV DX,0
00402021 │.  8A15 03104000   MOV DL,BYTE PTR DS:[401003]
00402027 │.  24 70           AND AL,70
00402029 │.  80E3 70         AND BL,70
0040202C │.  80E1 70         AND CL,70
0040202F │.  80E2 70         AND DL,70
00402032 │.  C0E8 04         SHR AL,4
00402035 │.  C0EB 04         SHR BL,4
00402038 │.  C0E9 04         SHR CL,4
0040203B │.  C0EA 04         SHR DL,4
0040203E │.  66:01D8         ADD AX,BX
00402041 │.  66:01C8         ADD AX,CX
00402044 │.  66:01D0         ADD AX,DX
00402047 │.  6A 00           PUSH 0                               ┌status = 0
00402049 └.  FF15 3C304000   CALL DWORD PTR DS:[<&msvcrt.exit>]   └MSVCRT.exit
0040204F  .  0000            ADD BYTE PTR DS:[EAX],AL
00402051  .  0000            ADD BYTE PTR DS:[EAX],AL
00402053  .  0000            ADD BYTE PTR DS:[EAX],AL
00402055  .  0000            ADD BYTE PTR DS:[EAX],AL
00402057  .  0000            ADD BYTE PTR DS:[EAX],AL
00402059  .  0000            ADD BYTE PTR DS:[EAX],AL
0040205B  .  0000            ADD BYTE PTR DS:[EAX],AL
0040205D  .  0000            ADD BYTE PTR DS:[EAX],AL
0040205F  .  0000            ADD BYTE PTR DS:[EAX],AL
00402061  .  0000            ADD BYTE PTR DS:[EAX],AL
00402063  .  0000            ADD BYTE PTR DS:[EAX],AL
00402065  .  0000            ADD BYTE PTR DS:[EAX],AL
00402067  .  0000            ADD BYTE PTR DS:[EAX],AL
00402069  .  0000            ADD BYTE PTR DS:[EAX],AL
0040206B  .  0000            ADD BYTE PTR DS:[EAX],AL
0040206D  .  0000            ADD BYTE PTR DS:[EAX],AL
0040206F  .  0000            ADD BYTE PTR DS:[EAX],AL
00402071  .  0000            ADD BYTE PTR DS:[EAX],AL
00402073  .  0000            ADD BYTE PTR DS:[EAX],AL
00402075  .  0000            ADD BYTE PTR DS:[EAX],AL

Stack [0019FF74]=0
Imm=0

problema13.<ModuleEntryPoint>+47
```

```
Registers (MMX)
EAX 00190011
ECX 00400006 problema13.00400006
EDX 00400002 problema13.00400002
EBX 002F0002
ESP 0019FF78
EBP 0019FF84
ESI 00402000 problema13.<ModuleEntryPoint>
EDI 00402000 problema13.<ModuleEntryPoint>

EIP 00402047 problema13.00402047

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 1  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2F4000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr 000000BB ERROR_SEM_NOT_FOUND

EFL 00000216 (NO,NB,NE,A,NS,PE,GE,G)

MM0 0000 0000 0000 0000
MM1 0000 0000 0000 0000
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000

XMM0                     0.0                      0.0
XMM1                     0.0                      0.0
XMM2                     0.0                      0.0
XMM3                     0.0                      0.0
XMM4                     0.0                      0.0
XMM5                     0.0                      0.0
XMM6                     0.0                      0.0
XMM7                     0.0                      0.0
                                       P U O Z D I
MXCSR 00001F80  FZ 0 DZ 0  Err  0 0 0 0 0 0
                Rnd NEAR   Mask 1 1 1 1 1 1
```

Modify EAX

	EAX	AX	AH	AL
Hexadecimal	00190011	0011	00	11
Signed	1638417	17	0	17
Unsigned	1638417	17	0	17
Character			\x00	\x11

```
Address  Hex dump                                          ASCII
00401000 7A 2F 61 2E 00 00 00 00 00 00 00 00 00 00 00 00  z/a.
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
004011B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004011C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
0019FF78 [76D26739 9gÒv RETURN to KERNEL32.BaseThreadInitThunk+19
0019FF7C  002F1000  /
0019FF80  76D26720  gÒv KERNEL32.BaseThreadInitThunk
0019FF84  0019FFDC Üÿ
0019FF88  77D28AFF ÿŠÒw RETURN to ntdll.77D28AFF
0019FF8C  002F1000  /
0019FF90  A4E96EC5 Åné¤
0019FF94  00000000
0019FF98  00000000
0019FF9C  002F1000  /
0019FFA0  00000000
0019FFA4  00000000
0019FFA8  00000000
0019FFAC  00000000
0019FFB0  00000000
0019FFB4  00000000
0019FFB8  00000000
0019FFBC  00000000
0019FFC0  00000000
0019FFC4  0019FF90  ÿ
0019FFC8  00000000
0019FFCC  0019FFE4 äÿ   Pointer to next SEH record
0019FFD0  77D3CD30 0ÍÒw SE handler
0019FFD4  D32C7B69 i{,Ó
0019FFD8  00000000
0019FFDC  0019FFEC ìÿ
0019FFE0  77D28ACD ÍŠÒw RETURN from ntdll.77D28AD4 to ntdll.77D28ACD
0019FFE4  FFFFFFFF ÿÿÿÿ End of SEH chain
0019FFE8  77D58725 %‡Òw SE handler
0019FFEC  00000000
```

Paused

Problema 24

Se da dublucuvantul M. Sa se obtina dublucuvantul MNew astfel:

bitii 0-3 a lui MNew sunt identici cu bitii 5-8 a lui M

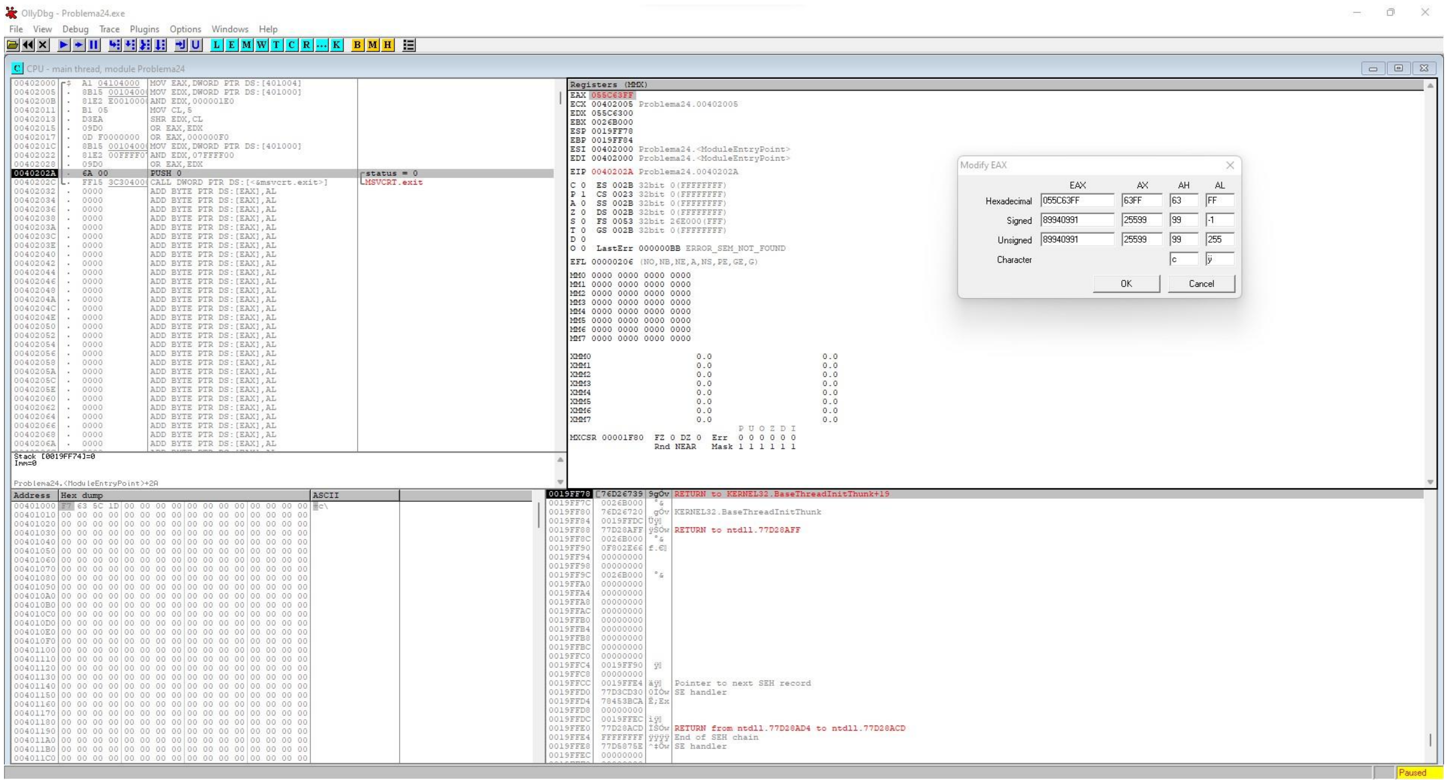
bitii 4-7 a lui MNew au valoarea 1

bitii 27-31 a lui MNew au valoarea 0

bitii 8-26 din MNew sunt identici cu bitii 8-26 a lui M.

```
bits 32 ; assembling for the 32 bits architecture


; declare the EntryPoint (a label defining the very first instruction of the program)

global start


; declare external functions needed by our program

extern exit               ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll    ; exit is a function that ends the calling process. It is defined in msvcrt.dll

                          ; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions


; our data is declared here (the variables needed by our program)

segment data use32 class=data

        M dd 1D5C63F7h ; 0001 1101 0101 1100 0110 0011 1111 0111b

        MNew dd 0

        ;Rezultat :      0000 0101 0101 1100 0110 0011 1111 1111b = 055C63FFh
```

```
; our code starts here
segment code use32 class=code
    start:
        MOV EAX, [MNew]
        MOV EDX, [M]
        AND EDX, 000001E0h   ; EDX = 0000 0000 0000 0000 0000 0001 1110 0000b
        MOV CL, 5
        SHR EDX, CL          ; EDX = 0000 0000 0000 0000 0000 0000 0000 1111b =0000000Fh
        OR EAX, EDX          ; EAX = 0000 0000 0000 0000 0000 0000 0000 1111b =0000000Fh

        OR EAX, 000000F0h;   ; EAX = 0000 0000 0000 0000 0000 0000 1111 1111b =000000FFh

        MOV EDX, [M]         ; Reinitializare DX

        AND EDX, 07FFFF00h   ; EDX = 0000 0101 0101 1100 0110 0011 0000 0000b
        OR EAX, EDX          ; EAX = 0000 0101 0101 1100 0110 0011 1111 1111b = 055C63FFh

        ;Rezultat final in AX
        ; exit(0)
        push    dword 0      ; push the parameter for exit onto the stack
        call    [exit]       ; call exit to terminate the program
```


OllyDbg - Problema24.exe
File View Debug Trace Plugins Options Windows Help
L E M W T C R ... K B M H
CPU - main thread, module Problema24
Modify EAX
OK
Cancel
Paused

```
00402000 ┌$  A1 04104000   MOV EAX,DWORD PTR DS:[401004]
00402005 │.  8B15 00104000 MOV EDX,DWORD PTR DS:[401000]
0040200B │.  81E2 E0010000 AND EDX,000001E0
00402011 │.  B1 05         MOV CL,5
00402013 │.  D3EA          SHR EDX,CL
00402015 │.  09D0          OR EAX,EDX
00402017 │.  0D F0000000   OR EAX,000000F0
0040201C │.  8B15 00104000 MOV EDX,DWORD PTR DS:[401000]
00402022 │.  81E2 00FFFF07 AND EDX,07FFFF00
00402028 │.  09D0          OR EAX,EDX
0040202A │.  6A 00         PUSH 0                                  ┌status = 0
0040202C └.  FF15 3C304000 CALL DWORD PTR DS:[<&msvcrt.exit>]      └MSVCRT.exit
00402032  .  0000          ADD BYTE PTR DS:[EAX],AL
00402034  .  0000          ADD BYTE PTR DS:[EAX],AL
00402036  .  0000          ADD BYTE PTR DS:[EAX],AL
00402038  .  0000          ADD BYTE PTR DS:[EAX],AL
0040203A  .  0000          ADD BYTE PTR DS:[EAX],AL
0040203C  .  0000          ADD BYTE PTR DS:[EAX],AL
0040203E  .  0000          ADD BYTE PTR DS:[EAX],AL
00402040  .  0000          ADD BYTE PTR DS:[EAX],AL
00402042  .  0000          ADD BYTE PTR DS:[EAX],AL
00402044  .  0000          ADD BYTE PTR DS:[EAX],AL
00402046  .  0000          ADD BYTE PTR DS:[EAX],AL
00402048  .  0000          ADD BYTE PTR DS:[EAX],AL
0040204A  .  0000          ADD BYTE PTR DS:[EAX],AL
0040204C  .  0000          ADD BYTE PTR DS:[EAX],AL
0040204E  .  0000          ADD BYTE PTR DS:[EAX],AL
00402050  .  0000          ADD BYTE PTR DS:[EAX],AL
00402052  .  0000          ADD BYTE PTR DS:[EAX],AL
00402054  .  0000          ADD BYTE PTR DS:[EAX],AL
00402056  .  0000          ADD BYTE PTR DS:[EAX],AL
00402058  .  0000          ADD BYTE PTR DS:[EAX],AL
0040205A  .  0000          ADD BYTE PTR DS:[EAX],AL
0040205C  .  0000          ADD BYTE PTR DS:[EAX],AL
0040205E  .  0000          ADD BYTE PTR DS:[EAX],AL
00402060  .  0000          ADD BYTE PTR DS:[EAX],AL
00402062  .  0000          ADD BYTE PTR DS:[EAX],AL
00402064  .  0000          ADD BYTE PTR DS:[EAX],AL
00402066  .  0000          ADD BYTE PTR DS:[EAX],AL
00402068  .  0000          ADD BYTE PTR DS:[EAX],AL
0040206A  .  0000          ADD BYTE PTR DS:[EAX],AL

Stack [0019FF74]=0
Imm=0

Problema24.<ModuleEntryPoint>+2A
```

```
Registers (MMX)
EAX 055C63FF
ECX 00402005 Problema24.00402005
EDX 055C6300
EBX 0026B000
ESP 0019FF78
EBP 0019FF84
ESI 00402000 Problema24.<ModuleEntryPoint>
EDI 00402000 Problema24.<ModuleEntryPoint>

EIP 0040202A Problema24.0040202A

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 26E000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr 000000BB ERROR_SEM_NOT_FOUND

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

MM0 0000 0000 0000 0000
MM1 0000 0000 0000 0000
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000

XMM0                 0.0                   0.0
XMM1                 0.0                   0.0
XMM2                 0.0                   0.0
XMM3                 0.0                   0.0
XMM4                 0.0                   0.0
XMM5                 0.0                   0.0
XMM6                 0.0                   0.0
XMM7                 0.0                   0.0
                                P U O Z D I
MXCSR 00001F80  FZ 0 DZ 0  Err  0 0 0 0 0 0
                Rnd NEAR   Mask 1 1 1 1 1 1
```

	EAX	AX	AH	AL
Hexadecimal	055C63FF	63FF	63	FF
Signed	89940991	25599	99	-1
Unsigned	89940991	25599	99	255
Character			c	ÿ

```
Address  Hex dump                                         ASCII
00401000 F7 63 5C 1D 00 00 00 00 00 00 00 00 00 00 00 00  ÷c\
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
004011B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004011C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
0019FF78 [76D26739 9gÒv RETURN to KERNEL32.BaseThreadInitThunk+19
0019FF7C  0026B000  °&
0019FF80  76D26720  gÒv KERNEL32.BaseThreadInitThunk
0019FF84  0019FFDC Üÿ
0019FF88  77D28AFF ÿŠÒw RETURN to ntdll.77D28AFF
0019FF8C  0026B000  °&
0019FF90  0F802E66 f.€
0019FF94  00000000
0019FF98  00000000
0019FF9C  0026B000  °&
0019FFA0  00000000
0019FFA4  00000000
0019FFA8  00000000
0019FFAC  00000000
0019FFB0  00000000
0019FFB4  00000000
0019FFB8  00000000
0019FFBC  00000000
0019FFC0  00000000
0019FFC4  0019FF90  ÿ
0019FFC8  00000000
0019FFCC  0019FFE4 äÿ   Pointer to next SEH record
0019FFD0  77D3CD30 0ÍÓw SE handler
0019FFD4  78453BCA Ê;Ex
0019FFD8  00000000
0019FFDC  0019FFEC ìÿ
0019FFE0  77D28ACD ÍŠÒw RETURN from ntdll.77D28AD4 to ntdll.77D28ACD
0019FFE4  FFFFFFFF ÿÿÿÿ End of SEH chain
0019FFE8  77D5875E ^‡Òw SE handler
0019FFEC  00000000
```