Tema Laborator 5

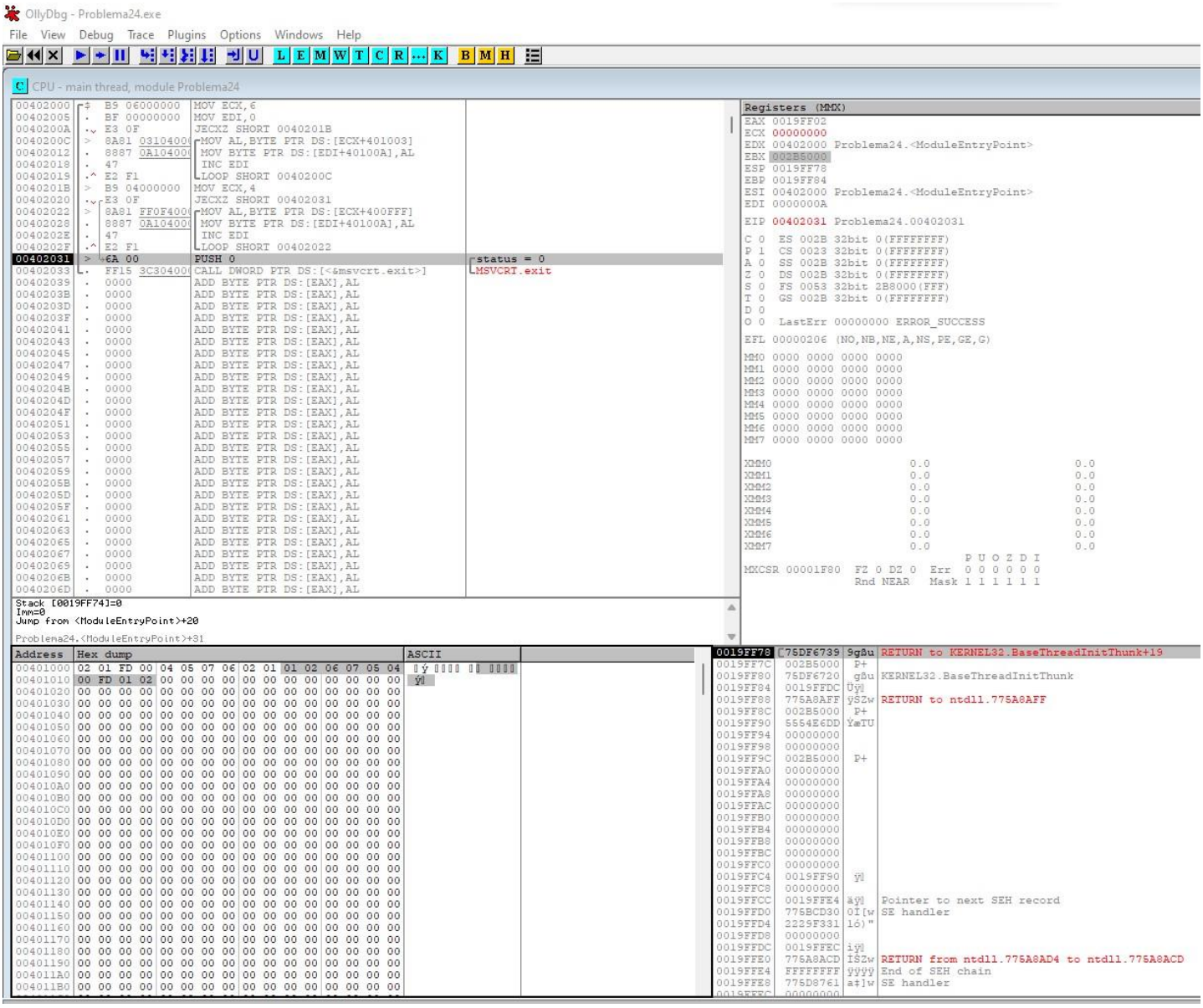
Suciu Radu – Grupa 216/2

Problema 24

Se dau 2 siruri de octeti A si B. Sa se construiasca sirul R care sa contina elementele lui B in ordine inversa urmate de elementele in ordine inversa ale lui A.

```
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 2, 1, -3, 0
    L1 EQU $-A
    B db 4, 5, 7, 6, 2, 1
    L2 EQU $-B
    R times L1+L2 db 0
    ;R: 1, 2, 6, 7, 5, 4, 0, -3, 1, 2
; our code starts here
segment code use32 class=code
    start:
        mov ECX, L2          ; L2 = 6
        mov EDI, 0           ;EDI = 0
        jecxz final1
        repeta:
            mov Al, [B+ECX-1]
            mov [R+EDI], Al
            inc EDI
        loop repeta
        final1:
        ; A doua instructiune LOOP pentru sirul A
          mov ECX, L1
        jecxz final2
        repeta2:
            mov AL, [A+ECX-1]
              mov [R+EDI], AL
```

```
        inc EDI
    loop repeta2
    final2:

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

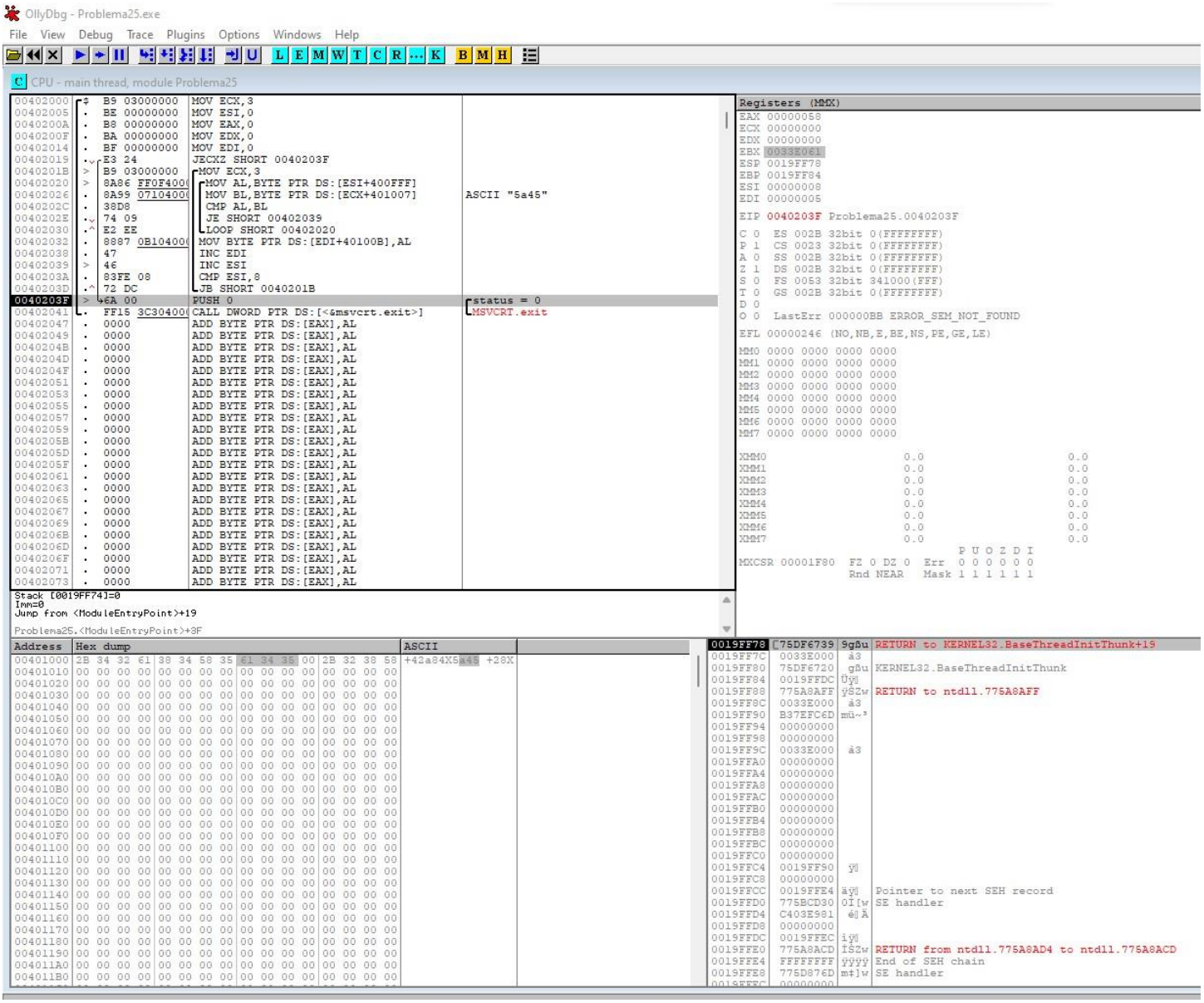


Problema 25

Se dau doua siruri de caractere S1 si S2. Sa se construiasca sirul D ce contine toate elementele din S1 care nu apar in S2.

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


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

```
extern exit

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

segment data use32 class=data

        S1 db '+', '4', '2', 'a', '8', '4', 'X', '5'

        L1 equ $-S1

        S2 db 'a', '4', '5'

        L2 equ $-S2

        D times L1 db 0

        ; our code starts here

segment code use32 class=code

    start:

        mov ecx, L2     ; L2 = 3

        mov esi, 0

        mov eax, 0

        mov edx, 0

        mov edi, 0

        jecxz final

        REPETA1:

            mov ecx, L2

            REPETA2:

                mov al, [S1+esi-1]  ; Se preiau elementele din siruri

                mov bl, [S2+ecx-1]  ;

                CMP al,bl           ; al - bl => rezultat in ZF

                jz eticheta         ;Conditia pentru elemente egale din siruri diferite

            LOOP REPETA2

            mov [D+edi], al         ; Elementul valid se adauga in D

            inc edi

            eticheta:

            inc esi                 ; ESI ++

            cmp esi, L1             ; ESI - L1

        JB REPETA1                  ;sare la eticheta REPETA1 daca CF = 1

        final:

        ; Rezultat final D: '+', '2', '8', 'X'

        push    dword 0      ; push the parameter for exit onto the stack

        call    [exit]       ; call exit to terminate the program
```


OllyDbg - Problema25.exe
File View Debug Trace Plugins Options Windows Help
CPU - main thread, module Problema25
00402000 $ B9 03000000 MOV ECX,3
00402005 . BE 00000000 MOV ESI,0
0040200A . B8 00000000 MOV EAX,0
0040200F . BA 00000000 MOV EDX,0
00402014 . BF 00000000 MOV EDI,0
00402019 . E3 24 JECXZ SHORT 0040203F
0040201B > B9 03000000 MOV ECX,3
00402020 > 8A86 FF0F4000 MOV AL,BYTE PTR DS:[ESI+400FFF]
00402026 . 8A99 07104000 MOV BL,BYTE PTR DS:[ECX+401007] ASCII "5a45"
0040202C . 38D8 CMP AL,BL
0040202E . 74 09 JE SHORT 00402039
00402030 . E2 EE LOOP SHORT 00402020
00402032 . 8887 0B104000 MOV BYTE PTR DS:[EDI+40100B],AL
00402038 . 47 INC EDI
00402039 > 46 INC ESI
0040203A . 83FE 08 CMP ESI,8
0040203D . 72 DC JB SHORT 0040201B
0040203F > 6A 00 PUSH 0 status = 0
00402041 . FF15 3C304000 CALL DWORD PTR DS:[<&msvcrt.exit>] MSVCRT.exit
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
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
Stack [0019FF74]=0
Imm=0
Jump from <ModuleEntryPoint>+19
Problema25.<ModuleEntryPoint>+3F
Registers (MMX)
EAX 00000058
ECX 00000000
EDX 00000000
EBX 0033E061
ESP 0019FF78
EBP 0019FF84
ESI 00000008
EDI 00000005
EIP 0040203F Problema25.0040203F
C 0 ES 002B 32bit 0(FFFFFFFF)
P 1 CS 0023 32bit 0(FFFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFFF)
Z 1 DS 002B 32bit 0(FFFFFFFF)
S 0 FS 0053 32bit 341000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
O 0 LastErr 000000BB ERROR_SEM_NOT_FOUND
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
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
Address Hex dump ASCII
00401000 2B 34 32 61 38 34 58 35 61 34 35 00 2B 32 38 58 +42a84X5a45 +28X
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
0019FF78 75DF6739 9gßu RETURN to KERNEL32.BaseThreadInitThunk+19
0019FF7C 0033E000 à3
0019FF80 75DF6720 gßu KERNEL32.BaseThreadInitThunk
0019FF84 0019FFDC Üÿ
0019FF88 775A8AFF ÿŠZw RETURN to ntdll.775A8AFF
0019FF8C 0033E000 à3
0019FF90 B37EFC6D mü~³
0019FF94 00000000
0019FF98 00000000
0019FF9C 0033E000 à3
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 775BCD30 0Í[w SE handler
0019FFD4 C403E981 é Ä
0019FFD8 00000000
0019FFDC 0019FFEC ìÿ
0019FFE0 775A8ACD ÍŠZw RETURN from ntdll.775A8AD4 to ntdll.775A8ACD
0019FFE4 FFFFFFFF ÿÿÿÿ End of SEH chain
0019FFE8 775D876D m‡]w SE handler