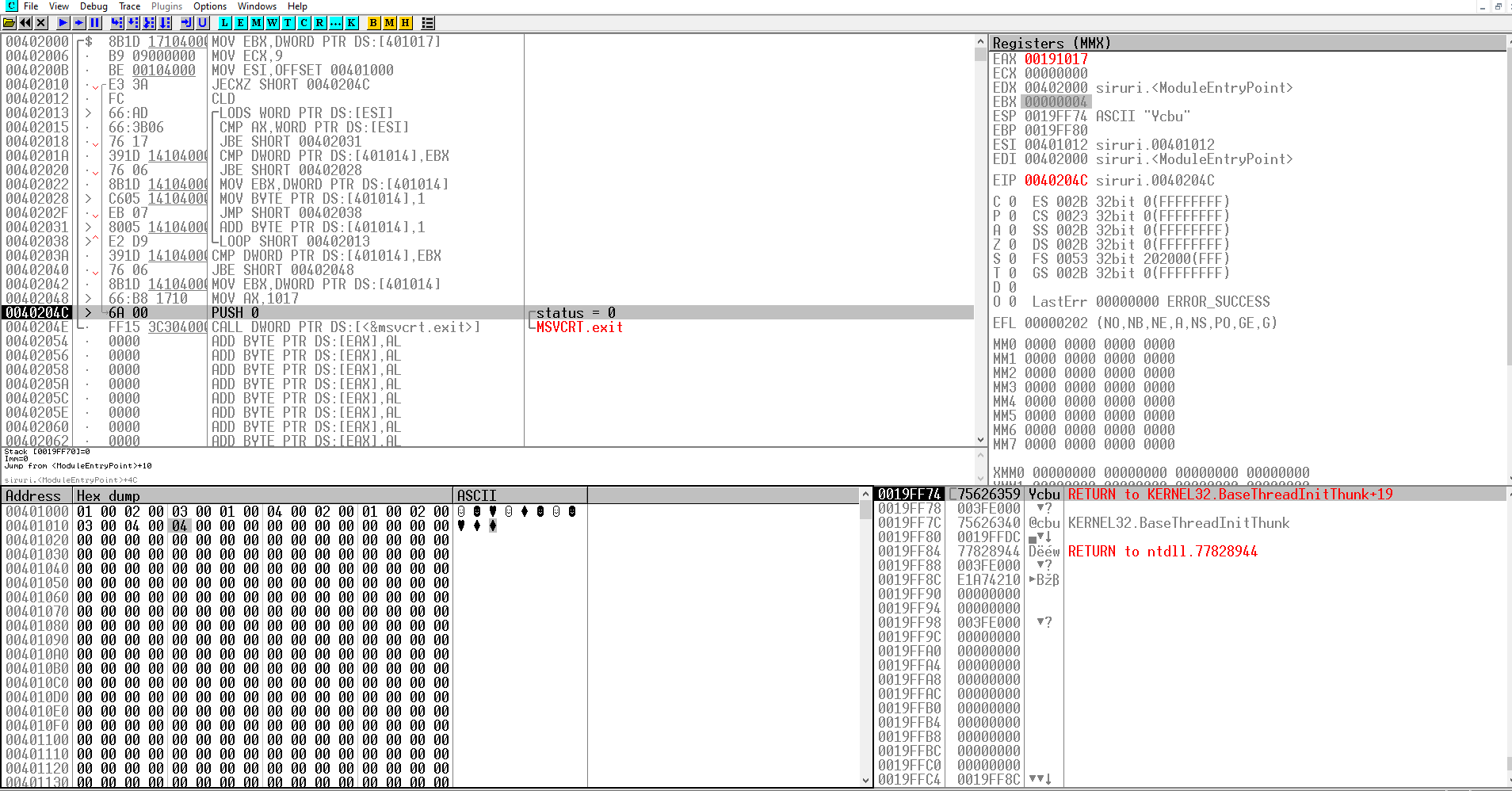
Tema laborator 6

Pop Mihai-Daniel, Grupa 215/2

Problema nr. 27: Varianta 1: st->dr (cld)

27.Dandu-se un sir de cuvinte, sa se calculeze cel mai lung subsir de cuvinte ordonate crescator din acest sir.

;sir = [1,2,3,1,4,2,1,2,3,4]

; 0,1,2,3,4,5,6,7,8,9

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

sir dw 1,2,3,1,4,2,1,2,3,4 ;sirul de cuvinte

len equ ($-sir)/2 ;lungimea sirului(in cuvinte)

contor db 1 ;contorul subsirului curent

subsir\_maxim db 0 ;lungimea subsirului de elemente ordonare crescator

rezultat dd 0 ;variabila pentru retinearea rezultatului

segment code use32 class=code ; segmentul de cod

start:

mov ebx, [subsir\_maxim] ;retinem in ebx lungimea subsirului maxim

mov ecx,len-1 ;punem in ecx lungimea sirului initial/ numarul de pasi

mov esi, sir ;sirul sursa

jecxz Sfarsit ;sare la final daca lungimea sirului este 0

cld ;parcurgem sirul de la stanga la dreapta (normal)

Repeta: ;inceputul buclei

lodsw ;in ax vom avea cuvantul curent din sir

cmp ax, [esi] ;comparam elementul salvat in ax cu urmatorul element

jbe CresteContor ;daca sunt in ordine crescatoare sare la incrementarea contorului

cmp [contor],ebx ;compara lungimea subsirului curent cu lungimea subsirului maxim

jbe Next0 ;sare la reinitializarea contorului daca contorul este mai mic decat lungimea maxima

mov ebx, [contor] ;altfel punem in ebx lungimea maxima

Next0:

mov byte[contor], 1 ;reinitializarea contorului

jmp Next ;sare la finalul buclei pentru a evita incrementarea incorecta a contorului

CresteContor:

add byte[contor],1 ;creste contorul

Next:

loop Repeta ;repetarea buclei

cmp [contor],ebx ;caz particular in care ultimul subsir este subsirul de lung maxima

jbe Next2

mov ebx, [contor] ;punem in ebx lungimea maxima

Next2:

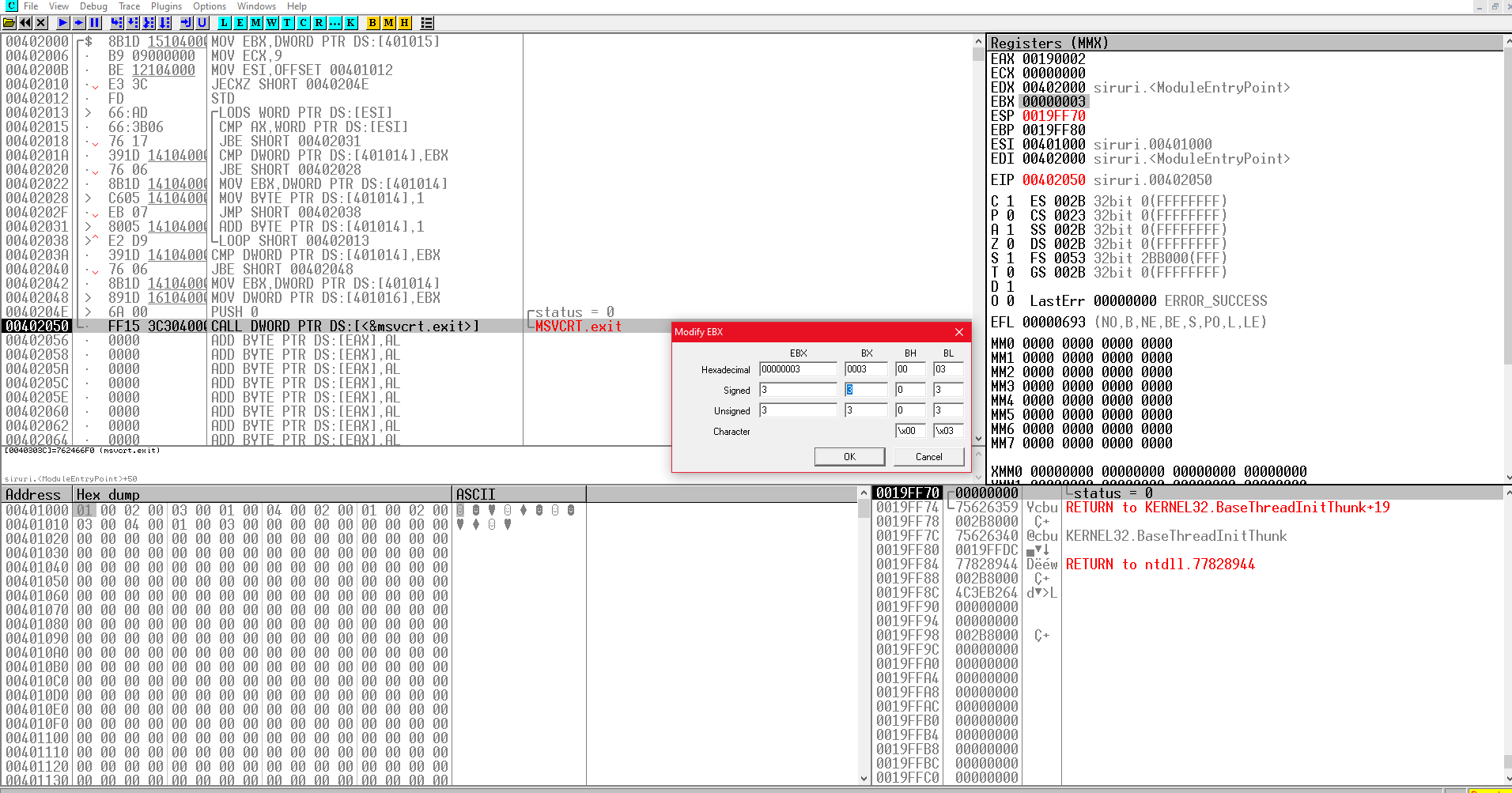
mov dword[rezultat],ebx ;punem in rezultat lungimea subsirului maxim

Sfarsit: ;sfarsitul programului

push dword 0

call [exit]

Varianta 2: st<-dr (std)



;27.Dandu-se un sir de cuvinte, sa se calculeze cel mai lung subsir de cuvinte ordonate crescator din acest sir.

;sir = [1,2,3,1,4,2,1,2,3,4]

; 9,8,7,6,5,4,3,2,1,0

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

sir dw 1,2,3,1,4,2,1,2,3,4 ;sirul de cuvinte

len equ ($-sir)/2 ;lungimea sirului(in cuvinte)

;destinatie times len dw 0 ;rezervam in memorie un numar de lun de cuvinte

contor db 1 ;contorul subsirului curent

;poz\_initiala\_subsir dw 0

subsir\_maxim db 0 ;lungimea subsirului de elemente ordonare crescator

rezultat dd 0 ;variabila pentru retinearea rezultatului

segment code use32 class=code ; segmentul de cod

start:

mov ebx, [subsir\_maxim] ;retinem in ebx lungimea subsirului maxim

mov ecx,len-1 ;punem in ecx lungimea sirului initial/ numarul de pasi

mov esi, sir+len\*2-2 ;sirul sursa

;mov edi, destinatie ;sirul destinatie

jecxz Sfarsit ;sare la final daca lungimea sirului este 0

std ;parcurgem sirul de la dreapta la stanga (invers)

Repeta: ;inceputul buclei

lodsw ;in ax vom avea cuvantul curent din sir

cmp ax, [esi] ;comparam elementul salvat in ax cu urmatorul element

jbe CresteContor ;daca sunt in ordine crescatoare sare la incrementarea contorului

cmp [contor],ebx ;compara lungimea subsirului curent cu lungimea subsirului maxim

jbe Next0 ;sare la reinitializarea contorului daca contorul este mai mic decat lungimea maxima

;mov poz\_initiala\_subsir, esi-contor\*2

mov ebx, [contor] ;altfel punem in ebx lungimea maxima

Next0:

mov byte[contor], 1 ;reinitializarea contorului

jmp Next ;sare la finalul buclei pentru a evita incrementarea incorecta a contorului

CresteContor:

add byte[contor],1 ;creste contorul

Next:

loop Repeta ;repetarea buclei

cmp [contor],ebx ;caz particular in care ultimul subsir este subsirul de lung maxima

jbe Next2

mov ebx, [contor] ;punem in ebx lungimea maxima

Next2:

mov dword[rezultat],ebx ;punem in rezultat lungimea subsirului maxim

Sfarsit: ;sfarsitul programului

push dword 0

call [exit]