ASC - 25 ian. 2019

II. +0la1 db '256, -256'

Se ia ca șir de caractere:

'2' | '5' | '6' | ',' | '-' | '2' | '5' | '6'

→ 8|a2 dw 256, 256h

256 = 100h

în memorie: 00|01|56|02

→ 12|a3 dw $-a2

$-a2 = 12 - 8 = 4 (scalar)

în memorie: 04|00

→ 14|a4 ogu -256|4

a4 este o constantă, deci nu se stochează în memorie, iar offset-ul nu crește

→ 14|a5 db 128 >> 1, -128 << 1

128 >> 1 = 1000 0000 >> 1 = 0100 0000 = 40h

-128 << 1 = C2(128) << 1 = 1000 0000 << 1 = 0000 0000 = 0h

în memorie: 40|00

→ 16|a6 dw a2-a5, ~(a2-a5)

a2-a5 = 8 - 14 = -6 = C2(0000 0000 0000 0110) =

scalar! = 1111 1111 1111 1010 = FFFAh

~(a2-a5) = 0000 0000 0000 0101 = 0009h

în memorie: FA|FF|09|00

→ a7 dd [a2], !a2

Syntax error, deoarece [a2] nu poate fi determinat la momentul asamblării, iar operatorul ! nu se poate aplica pe un pointer.

→ 20|a8 dd 256h ^256, 256256h

256h ^256 = 0010 0101 0110 ^ 0001 0000 0000 = 0011 0101 0110h

= 356h

in memorie 56|03|00|00 | 56|62|25|00

→ 28|a9 dd ($-a8)+(a10-$)

($-a8)+(a10-$) = (28-20)+(32-28) = 8+4 = 12 = 0ch

in memorie: 0c|00|00|00

→ 32|a10 dw -255, 256

-255 = C2(255) = 01h

256 = 100h

in memorie: 01|00|00|01

→ 36|a11 dw 256-256h

256-256h = 256-598 = -342 = C2(342) = EAAh

in memorie: AA|FE

→ 38|a12 times 4 dw 256

256 = 100h

in memorie 00|01|00|01|00|01|00|01

→ a13 dw times 4 -128

Syntax error

→ a14 dw -256

in memorie: 00|FF

→ a15 times 2 dd 12345678h

in memorie: 78|56|34|12 | 78|56|34|12

III. a) 1) lea ebx, [ebx+6] ; adună 6 la EBX și mută rez. în EBX

2) lea ebx, [bx+6] ; adună 6 la BX și mută rez în EBX

3) lea bx, [bx+6] ; adună 6 la BX și mută rez în BX

4) lea bx, [ebx+6] ; adună 6 la EBX și mută rez în BX

5) mov ebx, ebx+6 ; syntax error, valid doar în calculul offsetului sau dacă ar fi fost lea ebx, [ebx+6]

6) mov ebx, [ebx+6] ; pune în ebx valoarea de la adresa [ebx+6]

7) movzx ebx, [ebx+6] ; syntax error, trebuia menționată dimensiunea

8) movzx ebx, [bx+6] ; syntax error, —— // ——

9) add bx, 6 ; movzx ebx, bx ; adună 6 la bx ; a doua instr. nu face nimic

10) mov [ebx], dword [bx+6] ; syntax error, nu se pot aduce două valori din memorie

11) add ebx, 6 ; adună 6 la valoarea din EBX

12) add bx, 6 ; adună 6 la valoarea din BX

13) push [ebx+6] ; pop ebx ; syntax error (dim. nespecificată)

14) xchg ebx, [ebx+6] ; pune în EBX valoarea de la ebx+6 dacă este validă

Categoria 1: instrucțiuni care au același efect asupra registrului EBX sunt 1, 3, 4, 9, 11, 12

Categoria 2: instrucțiuni care semnalează erori de sintaxă și nu au nici un efect asupra registrului EBX: 5, 7, 8, 10, 13

b)
```
push ebp ; pune EBP pe stivă
mov ebp,esp ; și salvează în EBP (baza stivei) adresa vârfului stivei (ESP)
mov ebp, [ebp] ; în EBP se salvează valoarea de la adresa EBP (adică EBP de la început)
mov ebx, [ebp] ; în EBX se salvează valoarea de la adresa EBP
mov [esp], ebx ; în vârful stivei pune adresa lui EBX
               ; în ESP se pune valoarea de la adresa ebp
```