

Voorbeeld opgave week 2

Instructies

Dit is een voorbeeld opgave. Deze hoeft niet ingeleverd te worden.

Een uitwerking van deze opgave is beschikbaar. !

Voorbeeld 2a

We gebruiken in deze opgave de gecompileerde versie van **vb2a.c**.

1.	55	push	rbp
2.	48 89 e5	mov	rbp, rsp
3.	41 55	push	r13
4.	41 54	push	r12
5.	53	push	rbx
6.	bb 00 00 00 00	mov	ebx, 0x0
7.	41 bc 67 00 00 00	mov	r12d, 0x67
8.	41 bd dd 00 00 00	mov	r13d, 0xdd
9.	44 89 eb	mov	ebx, r13d
10.	45 89 e5	mov	r13d, r12d
11.	41 89 dc	mov	r12d, ebx
12.	c1 e3 02	shl	ebx, 0x2
13.	89 d8	mov	eax, ebx
14.	5b	pop	rbx
15.	41 5c	pop	r12
16.	41 5d	pop	r13
17.	5d	pop	rbp
18.	c3	ret	

De regels in **rood** hebben te maken met de proloog en epiloog van de methode (zie stack.pdf).

We bestuderen hier alleen de regels in **blauw**.

Actie 1

Op regel 3, 4 en 5 zien we een aantal push acties, en op regel 14,15 en 16 pop acties met dezelfde registers.

Vraag 1 : waarvoor dienen deze push en pop acties?

Vraag 1b : waarom staan de pop acties (r13,r12,rbx) in omgekeerde volgorde van de push acties (rbx,r12,r13) ?

Actie 2

We kijken nu alleen naar de regels 6 t/m 13.

Vraag 2 : geef bij elke regel de inhoud van alle relevante registers na de actie.
Gebruik hiervoor een tabel.

Actie 3

Vraag 3 : wat is de return waarde van dit programma?

Vraag 3b : Komt dit overeen met de waarde in EAX ?