Problemas 3 AC

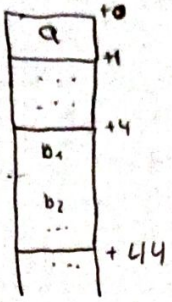
Problemas: 2.9, 2.10, 2.14

size int
|
4 × 10 = 40

2.9

s → %ebx
i → %esi
j → %edi
x → %dl

a)



b)

$s[i] = @s + 4 \cdot i$
$b[j] = @b + j \cdot 4$
} És un struct

$s[i].b[j] = @s + 44 \cdot i + 4 + j * 4$

(vect b)
(struct que toca)

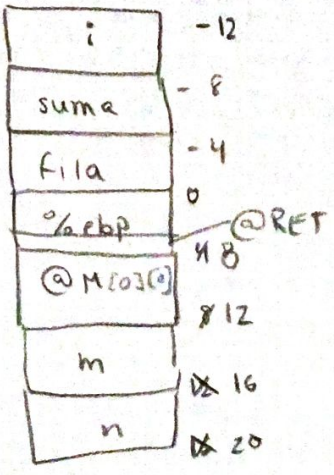
c)

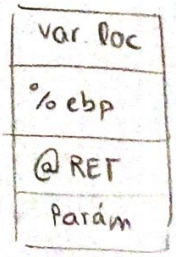
```
mull $44, %esi, %eax      # i · 44
addl %ebx, %eax           # res = @s + i · 44
addl $4, %eax             # res += 4
addl (%eax, %edi, 4), %eax   # @s[i].b[j]
imull (%eax, %ebx) $44, %eax, %eax   # s[s[i].b[j]] # 4
                          # @s + 44 · eax
addl %ebx, (%eax)4,
addl $1, %eax
movl %eax, %dl
```

%eax = devolver resultados

2.10

a) var. loc : 3 4 = 12





```
calcula: pushl %ebp
         movl %esp, %ebp
         subl $12, %esp
         movl $0, -8(%ebp)
         movl $0, -4(%ebp)
for:     movl 12(%ebp), %edx
         cmpl 16(%ebp), %edx
         jle fifor
         leal 8(%ebp), %eax
         addl %eax, (-4(%ebp), 40), %eax
         addl %edx, (%edx, 4), %edx
         leal -4(%ebp), %ebx
         call Normaliza
```

{¿qué hace esto?

suma = 0
fila = 0
i = m

LEAL = op2 ← &op1

2.14

a)

b)
```
                    # eax = 0
xorl %eax, %eax
leal -404(%ebp), %ebx
leal -4(%ebp), %ecx
pushl %eax
pushl %ebx
pushl %ecx
call examen
```

c)
```
    movl $0, -4(%ebp)
for: cmpl $100, -4(%ebp)
     jge fifor
     leal -404(%ebp), %eax
     addl %eax, -4(%ebp, 4), %eax
     addl 12(%ebp), -4(%ebp, 4), %ebx
     movl %eax, %ebx
     incl -4(%ebp)
     jmp for
fifor:
```

d)
```
movl 8(%ebp), %eax
movl 12(%ebp), %ebx
movl 16(%ebp), %ecx
```