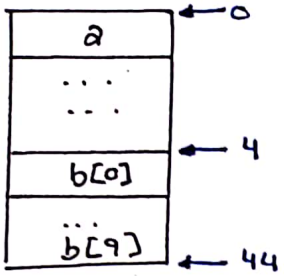
RICARD GUIARD

2.9. Structs:

A)



B) ⑤ + i·44 + 4 + 4·j

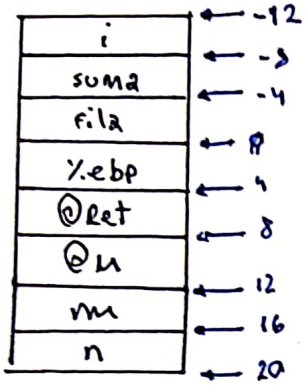
PRIMERA POS B

5 SENCER

C)
```
imull $44, %esi, %eax  # 44·i
addl %ebx, %eax
imull $44, 4(%eax,%edi,4), %eax
movb (%ebx,%eax), %dl
```

2.10. Subrutines:

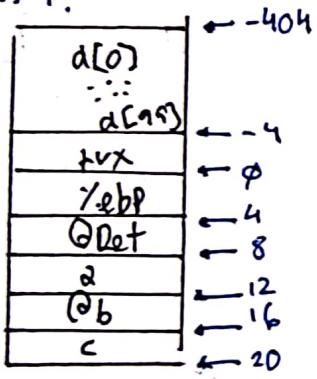
A)



B)
```
pushl %ebp
movl %esp, %ebp
subl $12, %esp
pushl %ebx
movl $0, -8(%ebp)
movl $0, -4(%ebp)
movl 12(%ebp), %ebx
For: cmpl 16(%ebp), %ebx
jge FiFor
...
leal -4(%ebp), %eax
push %eax
movl -4(%ebp), %edx
imull $10, %edx
addl %ebx %edx
movl 8(%ebp), %ecx
movl (%ecx, %edx, 4), %edx
pushl %edx
call normalitza
addl $8, %esp
...
addl %eax, -8(%ebp
incl %ebx
jmp For
FiFor: movl -8(%ebp), %eax
incl %eax
popl %ebx
movl %ebp, %esp
popl %ebp
ret
```

2.14.

A)



B)
```
examen(c, d, &aux):
leal -4(%ebp), %eax // &ebp-4 → %eax (aux)
leal -404(%ebp), %ecx // &ebp-404 → %ecx (d)
pushl %eax
pushl %ecx
pushl $0
call examen
```

C)
```
examen(a, b, c)
pushl 16(%ebp)
pushl 12(%ebp)
pushl 8(%ebp)
call examen
```

C)
```
For (aux=0; aux<100; aux++) b[aux] = d[aux];
movl $0, %ecx  #%ecx = 0 (aux)
For: cmpl $100, %ecx
jge FiFor
leal -404(%ebp), %eax
movl (%eax, %ecx, 4), %edx
movl 12(%ebp), %eax
movl %edx, (%edx, %ecx, 4)
incl %ecx
jmp For
FiFor:
```