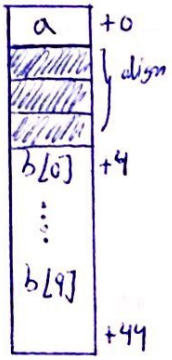
PROBLEMES AC:

2.9) $s \to$ %ebx, $i \to$ %esi, $j \to$ %edi, $x \to$ %dl

a) elem:



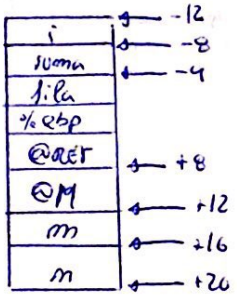
b) acces vector: @inicioV + i·tam

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

c) $x = s[s(i).b(j)].a$

```
imull $44, %esi, %eax   // eax → 44i
addl %ebx, %eax         // eax → eax + @s
imull $44, 4(%eax,%edi,4), %eax   // eax → 44·@[eax+j·4]+4]
movb (%ebx,%eax), %dl   // x → @[eax+ebx]
```

2.10) a) bloq activación



b) traducción:

```
calcula: pushl %ebp
         movl %esp, %ebp
         subl $12, %esp        // espai per var locals
         pushl %ebx            // %ebx a pila
         movl $0, -8(%ebp)     // suma = 0
         movl $0, -4(%ebp)     // 1ila = 0
         movl 12(%ebp), %ebx   // i = m
for:     cmpl 16(%ebp), %ebx   // i<m
         jge fi_for            // salto si i≥m
         leal -4(%ebp), %eax   // eax → @1ila
         pushl %eax            // eax a pila
         movl -4(%ebp), %eda
         imull $10, %edx       // 10·1ila
         addl %ebx, %edx       // 10·1ila + i
         movl 8(%ebp), %eax    // @M
         movl (%eax,%edx,4), %eda //
         pushl %edx            // edx a pila
         call Normalitza
```

*

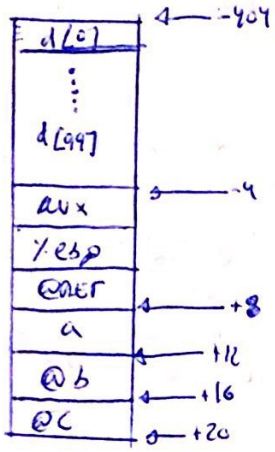
```
addl $8, %esp
addl %eax, -8(%ebp)
incl %ebx
jump for
```

```
fi_for: movl -8(%ebp), %eax   // @suma
        incl %eax             // suma+1
        popl %ebx             // allibera espai pila estat inicial
        movl %ebp, %esp
        popl %ebp
        ret.
```

*

2.14)

a) Bloq activació:



b) Traducción examen(a, d, &aux):

```
leal -4(%ebp), %eax    // eax ← &aux
leal -404(%ebp), %ecx  // ecx ← &d[0]
pushl %eax             // preparo per la crida
pushl %ecx
pushl $0
call examen
```

c) Traducció:

```
    movl $0, %ecx      // aux = 0
for: cmpl $100, %ecx   // aux < 100
    jge fi_for         // salto si aux ≥ 100
    leal -404(%ebp), %eax // eax ← &d[0]
    movl (%eax, %ecx, 4), %eax // eax ← d[aux]
    movl 12(%ebp), %edx  // edx ← &b
    movl %eax, (%edx, %ecx, 4) // b[aux] = d[aux]
    incl %ecx          // ++aux
    jmp for
fi_for:
```

d) Traducción: examen(a, b, c):

```
pushl 16(%ebp)
pushl 12(%ebp)
pushl 8(%ebp)
call examen
```