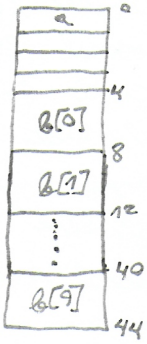
Sessió 3 ex. 9, 10, 14 ← Tema 2

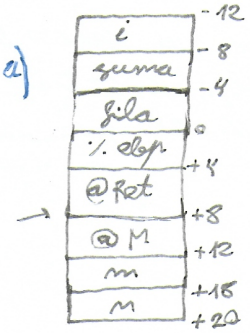
(9) typedef struct { s → %ebx, i → %esi, j → %edi, x → %dl
  char a;
  int b[10];
} elem;
elem s[100];

a)


b) @s[i].b[j] = @s + i·44 + j·4

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

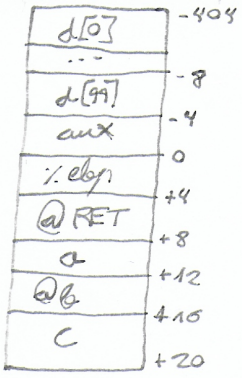
```
imull $44, %esi, %eax      # i*44
addl %ebx, %eax            # @s + i*44
imull $44, (%eax, %edi, 4), %eax   # 44*(@s + i*44)
movb (%ebx, %eax), %dl     # @s+
```

(10) a)


i → %ebx

b)
```
pushl %ebp
movl %esp, %ebp
subl $12, %esp          # %esp ← %esp − 12
pushl %ebx              # %esp ← %esp + 4
movl $0, -8(%ebp)       # suma = 0
movl $0, -4(%ebp)       # fila = 0
movl 12(%ebp), %ebx     # i = m
for: cmpl 16(%ebp), %ebx   # salta si i >= m
  jge endfor
  movl -4(%ebp), %eax      # %eax ← fila
  pushl %eax               # %esp ← %esp + 4
  movl -4(%ebp), %edx      # %edx ← fila
  imull $10, %edx          # %edx ← fila*10
  addl %ebx, %edx          # %edx ← fila*10 + i
  movl 8(%ebp), %ecx       # %ecx ← @M
  movl (%ecx, %edx, 4), %edx   # %edx ← @M + (fila*10+i)*4
  pushl %edx               # %esp ← %esp + 4
  call Normalitza
  addl $8, %esp
  addl %eax, -8(%ebp)      # suma += retorn de Normalitza
  incl %ebx
endfor: movl -8(%ebp), %eax   # %eax ← suma
  incl %eax                   # suma + 1
  popl %ebx
  movl %ebp, %esp
  popl %ebp
  ret
```

(14) a)



b) examen(0, d, &aux)

```
leal -4(%ebp), %eax
pushl %eax
leal -404(%ebp), %ecx
pushl %ecx
pushl $0
call examen
```

c) for (aux = 0, aux < 100; aux++)
b[aux] = d[aux];

```
movl -4(%ebp), %ecx
movl $0, %ecx      # i = 0
for: cmpl $100, %ecx   # salta si
jge endfor
leal -404(%ebp), %eax   # %eax ← @d
movl (%eax, %ecx, 4), %ebx   # %ebx ← @d+4*i
movl 12(%ebp), %edx   # %edx ← b
movl %ebx, (%edx, %ecx, 4)   # b[aux] = d[aux]
incl %ecx   # aux++
jmp for
endfor:
```

d) examen(a, b, c);

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