

# AC Problemas Sesión 4: 2.18, 2.19

2.18

```

int mat1[M][N];
int mat2[N][M];

int SumaElemento(int i, int j){
    return mat1[i][j] + mat2[i][j];
}
    
```

a)

SumaElemento

```

0. pushl %ebp
1. movl %esp, %ebp
2. movl 8(%ebp), %eax //eax ← i
3. movl 12(%ebp), %ecx //ecx ← j
4. sall $2, %ecx //ecx ← ecx * 4
5. leal (, %eax, 8), %edx //edx ← 8 * i
6. subl %eax, %edx //edx ← 8i - i = 7i
7. leal (%eax, %eax, 4), %eax //eax ← i * 4 = 4i
8. movl mat2(%ecx, %eax, 4), %eax //eax ← M[8mat2 + 4j + 4 * 5i] = M[8mat2 + 4j + 20i] = mat2[i][j]
9. addl movl mat1(%ecx, %eax, 4), %eax //eax ← M[8mat1 + 4j + 4 * 7i] = M[8mat1 + 4j + 28i] = mat1[i][j]
10. movl %ebp, %esp
11. popl %ebp
12. ret
    
```

$mat1[i][j] \Rightarrow 4 \cdot (i \cdot N + j)$   
 $mat2[i][j] \Rightarrow 4 \cdot (i \cdot M + j)$

$$4(i \cdot N + j) = 4iN + 4j = 20i + 4j \rightarrow M = \frac{20i}{4i} = 5$$

$$4(i \cdot N + j) = 4iN + 4j = 28i + 4j \rightarrow N = \frac{28i}{4i} = 7$$

b) Tiene 13 instrucciones estáticas

c) Tiene 13 instrucciones dinámicas (totales)

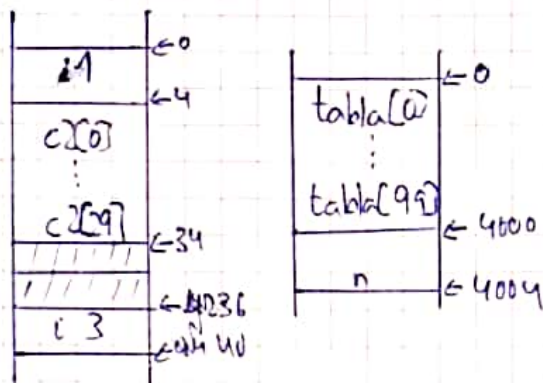
d) 4 accesos a Memoria (subtallado)

e)  $CPI_{\text{mem}} = \frac{1}{0.5}$   $CPI_{\text{instr}} = \frac{1}{0.5}$   $9 \text{ instr.} \left( \frac{1}{0.5} \right) + 4 \cdot \left( \frac{1}{0.5} \right) = 19.25 \text{ ciclos}$

f)  $CPI = \frac{1}{0.1} \rightarrow CPI_{\text{mem}} = 9 \cdot \left( \frac{1}{0.1} \right) + 4 \cdot \left( \frac{1}{0.1} \right) = 16.667 \text{ ciclos}$   $Speedup = \frac{19.25}{16.667} = 1.155 \text{ veces más rápido}$

2.19

a)



b)

i	-98
j	-94
aux	-40
%ebp	0
@RET	4
P1	8
x	12
y	16
	20

c) `movl 12(%ebp), %eax`  
`movl (%eax), %eax`  
`addl -4(%ebp), %eax`

d) `movl 8(%ebp), %eax`  
~~`movl 40`~~  
`movl -44(%ebp), %ebx`  
`imull $40, %ebx`  
`addl %ebx, %eax`  
`movl 16(%ebp), %ebx`  
`pushl %ebx`  
`pushl %eax`  
`call F`  
`addl $8, %esp`  
`movl %eax, -40(%ebp)`

e) `movl -44(%ebp), %eax`  
`movl 16(%ebp), %ebx`  
`imull %eax, %ebx`  
`movl %ebx, -48(%ebp)`

f) `movb -13(%ebp), %al`  
`leal -40(%ebp), %ebx`  
`addl $4, %ebx`  
`addl -48(%ebp), %ebx`  
`movb %al, (%ebx)`

g) `pushl %esi`  
`movl $0, %eax`  
`movl 8(%ebp), %ebx`

`for: cmpl 16(%ebp), %eax`  
`jge fi_for`  
`cmpl 4000(%ebp), %eax`  
`jge fi_for`  
`imull $40, %eax, %ebx`  
`addl %ebx, %ebx`  
~~`movl 36(%ebp), %esi`~~  
`movl 36(%ebp), %esi`  
`addl %eax, %esi`

`movl %eax, (%ebx)`  
`addl $5, %eax`  
`jmp for`  
`fi_for: popl %esi`



$$\text{'!'} = 46 = 0x2E \quad \text{'\#'} = 35 = 0x23$$

```

b) movl -40(%ebp), %eax
    cmpl 16(%ebp), %eax
    je else
    movl -48(%ebp), %ebx
    jmp end
else: movl 40
    movl -44(%ebp), %ebx
    jmp end
end: movl %ebx, -4(%ebp)

```

```

i) movl $0, %eax
    leal -40(%ebp), %ebx
while: cmpl 40(%ebp), %eax
while: cmpl $0x2E, 4(%ebx, %eax)
    je fi_while
    movb $0x23, 4(%ebx, %eax)
    incl %eax
    jmp while
fi_while:

```