

Nombre: TONI BOSCH CASTELLS

Grupo: 21

Nombre: PAU CARBÓ PASCUAL

### Hoja de respuesta al Estudio Previo

1. for (i=0, suma=0; i<N; i++)  
    suma+=Matriz[i][2];

La traducción a código ensamblador del anterior código C es:

```
movl $0, %eax #i
movl $0, %ebx #suma
leal Matriz, %ecx #@Matriz[0][0]
for: cmpl $10, %eax
     jge endfor
     addl 8(%ecx, %eax, 4), %ebx #Matriz[i][2]: %ecx + (10i + 2) * 4 = %ecx + 40i + 8
     incl %eax
     jmp  for
endifor:
```

2. Realizando acceso secuencial la traducción es:

```
movl $0, %ebx #suma
leal Matriz, %ecx #@Matriz[0][2]
leal Matriz+368, %ecx #@Matriz[9][2]
for: cmpl %eax, %ecx
     jg  endfor
     addl (%ecx), %ebx
     addl $40, %ecx
     jmp  for
endifor:
```

La versión aleatoria ejecuta:

55

instrucciones. La secuencial ejecuta:

55

instrucciones.

```
3. for (i=0, j=0, ResFila[0]=1; i<N; i++, j=0, ResFila[i]=1)
    while (Matriz[i][j] != 0) {
        if (Matriz[i][j] > M)
            ResFila[i] -= Matriz[i][j];
        j++;
    }
```

La traducción a código ensamblador del anterior código C es:

```
    movl    $0, %eax    # i
    leal    Matriz, %ecx  # @Matriz[0][0]
    leal    ResFila, %edx  # @ResFila[0]
    movl    $1, (%edx)
for:  cmpl   $10, %eax    # i < N
     jge    endfor
while: cmpl   $0, (%ecx)  # Matriz[i][j] != 0
     je     endwhile
     if: cmpl   $100, (%ecx) # Matriz[i][j] > M
     jle    else
     movl    (%ecx), %ebx
     subl    %ebx, (%edx, %eax, 4) # ResFila -= Matriz[i][j]
else:  addl    $4, %ecx
     jnp     while
endwhile: incl %eax
     leal    Matriz, %ecx
     movl    %eax, %edi
     imul    $40, %edi
     addl    %edi, %ecx    # @Matriz[i][j]
     movl    $1, (%edx, %eax, 4)
     jnp     For
endfor:
```