

Nombre:

DNI:

Grupo:

Sobre 10, cada respuesta vale 2 si es correcta, 0 si está en blanco o claramente tachada, y -2/3 si es errónea.
Anotar las respuestas (a, b, c o d) en la siguiente tabla.

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| a | b | d | b | b |

- En x86_64 se pueden referenciar los registros... p.202 libro (F.3.2), p.307-8 (F.3.35), Prác2 Ap.2 p.15, T2.4 tr.4, T2.1 tr.26,43
 - %rax, %eax, %ax, %ah, %al siempre que no vayan %ah y uno de los nuevos (p.ej. %sil) en la misma instrucc.
 - %rsi, %esi, %si, %sih, %sil %sih no existe
 - %r8, %r8d, %r8w, %r8l %r8l no existe, es %r8b
 - %r12q, %r12d, %r12w, %r12l ni %r12l ni %r12q, es %r12 a secas
- Comparando las convenciones de llamada de gcc Linux IA-32 con x86_64 respecto a registros... T2.3 tr.36, T2.4 tr.5
 - En IA-32 %ebx es salva-invocante, pero en x86_64 %rbx es salva-invocado ambos salva-invocado
 - En IA-32 %ecx es salva-invocante, y en x86_64 %rcx es salva-invocante también ok
 - En IA-32 %esi es salva-invocado, y en x86_64 %rsi es salva-invocado también en x86_64 salva-invocante
 - En IA-32 %ebp es especial (marco de pila), y en x86_64 %rbp también en x86_64 salva-invocado
- Si definimos `int vec_ele(int n, int *a, int i, int j){ return a[n*i+j];}`, entonces...
 - Se puede llamar a la función con cualquier vector a que tenga al menos $n*i+j$ elementos no, $(n*i+j)+1$ elementos
 - Se espera que a sea un array bidimensional de n filas y n columnas
 - Se espera que a sea un array bidimensional de n filas, i es la fila y j la columna
 - Se espera que a sea un array bidimensional de n columnas ok, T2.4 tr.32
- Una función C llamada `get_el()` genera el siguiente código ensamblador. Se puede adivinar que...


```
movl 8(%ebp), %eax
movl var(,%eax,4), %edx
movl 12(%ebp), %eax
movl (%edx,%eax,4), %eax
```

 - var es un array multi-nivel (punteros a enteros) de cuatro filas
 - var es un array multi-nivel pero no se pueden adivinar las dimensiones ok, T2.4 tr.30
 - var es un array bidimensional de enteros, no se pueden adivinar dimensiones
 - var es un array bidimensional de enteros, con cinco columnas
- Al traducir la sentencia C `return &r->a[idx];` gcc genera el código ASM siguiente. Se deduce que...


```
movl 12(%ebp), %eax
sall $2, %eax
addl 8(%ebp), %eax
```

 - r es una estructura, y su dirección (&r) es 12(%ebp)
 - a es un array y es el primer campo de la estructura ok, T2.4 tr.41
 - idx vale 12
 - idx vale 4