

**Problema 9. (1) Structs**

Dada la siguiente declaración de variables:

```
typedef struct{
    char a;
    int b[10];
} elem;
elem s[100];
```

Sabemos que la dirección de `s` se encuentra en `%ebx` y las variables `i`, `j` y `x` se encuentran respectivamente en los registros `%esi`, `%edi` y `%dl`

- a) Dibujad el struct `elem`.
- b) Determinad cómo se calcula la dirección de memoria donde se almacena el dato  
`s[i].b[j]`
- c) Escribid la secuencia de instrucciones necesaria para codificar la siguiente sentencia  
`x = s[s[i].b[j]].a`

**Problema 10. (1) Subrutinas**

Dada la siguiente función escrita en C:

```
int calcula(int M[10][10], int m, int n)
{
    int i, suma, fila;
    suma = 0;
    fila = 0;
    for (i = m; i < n; i++)
        suma = suma + Normaliza(M[fila][i], &fila);
    return (suma + 1);
}
```

- a) Dibujad el bloque de activación de la función.
- b) Traducid la función a ensamblador del IA32. Suponed que la función **Normaliza** ya está programada.

**Problema 14.**

Dada la siguiente función escrita en C:

```
void examen (int a, int b[100], int *c)
{
    int d[100];
    int aux;
    ...
}
```

- a) Dibujad el bloque de activación de la función.

Traducid a ensamblador del IA32 las siguientes sentencias suponiendo que se encuentran en el cuerpo de la rutina **examen**:

- b) Traducid a ensamblador IA32 la siguiente sentencia suponiendo que se encuentra en el cuerpo de la rutina **examen**:
- ```
examen(0, d, &aux);
```
- c) Traducid a ensamblador IA32 la siguiente sentencia suponiendo que se encuentra en el cuerpo de la rutina
- ```
for (aux = 0; aux < 100; aux++)
    b[aux] = d[aux];
```
- d) Traducid a ensamblador IA32 la siguiente sentencia suponiendo que se encuentra en el cuerpo de la rutina
- ```
examen(a, b, c);
```