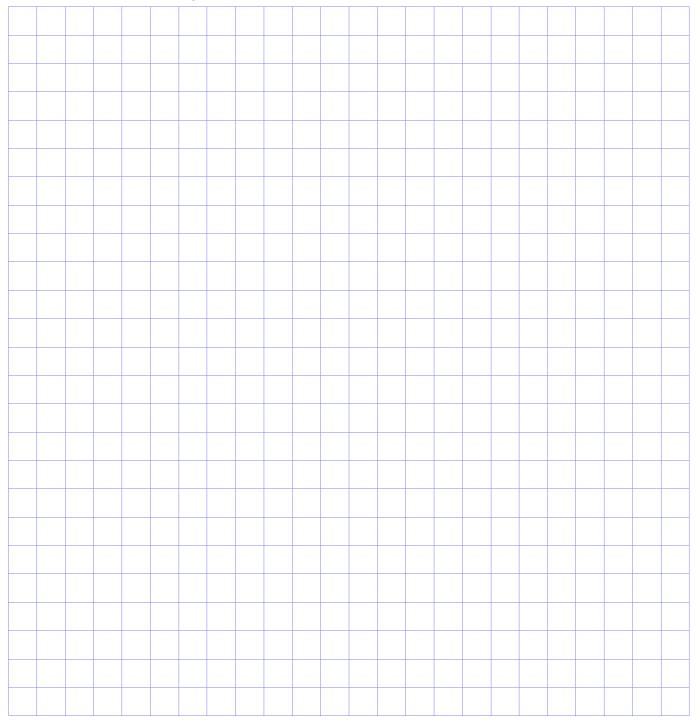
# 1er Control Estructura de Computadores II

## Problema 1 (2 puntos)

Dada la siguiente rutina en C:

```
int rutina(int *j) {
  int k;
  k = *j-2;
  if (k>32)
    k = rutina(&k);
  return (k);
}
```

Traducid literalmente el código a ensamblador.



curso 2007-2008 (Q1) 1/6

#### Problema 2 (2 puntos)

Dadas las siguientes estructuras en C:

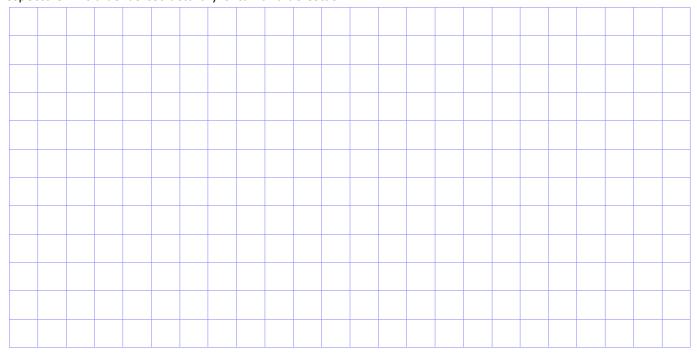
```
typedef struct {
  char c1;
  int a;
  char c2;
  char *p;
} ss;

big *ps; /* almacenado en %ebx */
int i; /* almacenado en %edi */

int j; /* almacenado en %edi */
```

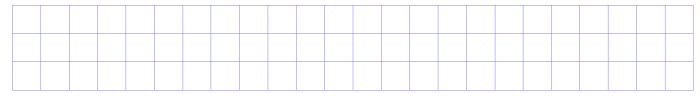
Suponiendo que ps se encuentra en %ebx, i en %esi, y j en %edi, responded a las siguientes preguntas:

a) Dibujad como quedarían almacenadas las estructuras de datos en linux, indicando claramente los desplazamientos respecto el inicio de las estructura y el tamaño de éstas.



b) Escribid una secuencia de 2 instrucciones que realice la siguiente asignación:

ps->v[4]->a++;



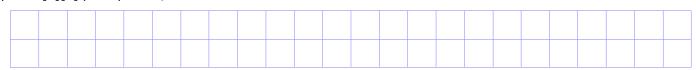
c) Escribid una secuencia de instrucciones que realice la siguiente asignación (se puede hacer en 5):

ps->m[i][j].c1 = ps->m[i][j].c2;



d) Escribid 1 instrucción que realice la siguiente asignación:

ps->m[2][1].p = &ps->c3;



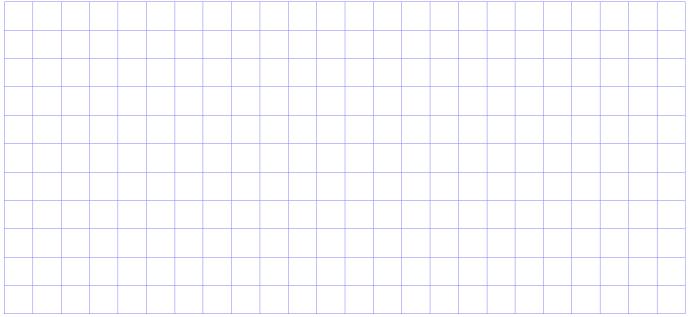
curso 2007-2008 (Q1) 2/6

#### Problema 3 (2 puntos)

Dado el siguiente código escrito en C (ss es el mismo struct usado en el problema 2):

```
int ec2(ss X, int *j, int i) {
  char v[23];
  int tmp;
  ...
  (1) while (tmp<23) && (i>10) {
    v[tmp] = v[tmp] + 33;
    tmp++;
    i--;
    }
  ...
  (2) return (*j + (int)X.c2);
}
```

a) **Dibujad** el bloque de activación de la función ec2.



b) **Traducid** literalmente a ensamblador de IA32 la sentencia (1).



c) **Traducid** literalmente a ensamblador de IA32 la sentencia (2).



curso 2007-2008 (Q1) 3/6

### Pregunta 4 (2 puntos)

1) Dado el siguiente formato de punto flotante:

S	Exponente		Mantisa			
13	12		9	8		0

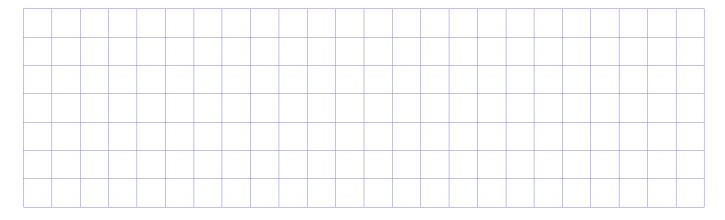
Donde: S es el bit de signo, el exponente se representa en exceso 8, hay bit implícito y la mantisa está normalizada.

Representad en este formato el número 15,65. En caso necesario redondead por truncamiento.



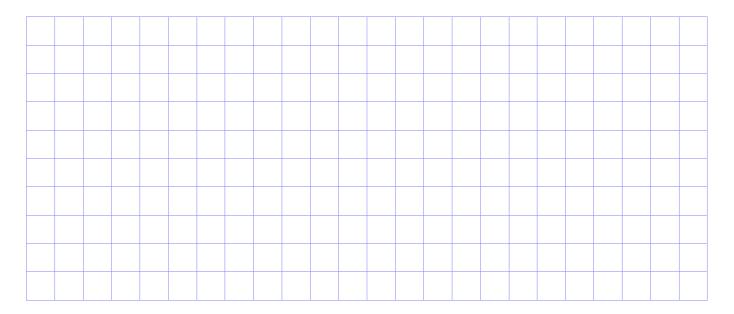
2) **Escribid** una secuencia de microoperaciones que se comporte de la misma forma que la siguiente instrucción IA32: **popl -9(, %eax,4)**. Las microoperaciones que podéis utilizar son las siguientes:

```
Ri = Rj op Rk
Ri = GetDesplazamiento(IR)
Ri = M[Rj]
Ri = GetFactorEscala(IR)
M[Rj] = Ri
Ri = GetInmediato(IR)
# siendo op: +, -, >>, <<, *, /</pre>
```



3) **Escribid** una secuencia de instrucciones para ejecutar la operación a = ((a\*b)+c)/(a-d) en una arquitectura de tipo **acumulador** en la que las operaciones se describen de la sigiente forma:

add/sub/mul/div @ #ACC = ACC op M[@]
load @ #ACC = M[@]
store @ #M[@] = ACC



curso 2007-2008 (Q1) 4/6

4) **Escribid** la secuencia de comandos Linux necesaria para extraer los ficheros contenidos en "Programas.Sesion05.tar.gz".



5) Suponiendo que a partir de la posición de memoria 0x98654E se encuentra almacenada la siguiente secuencia de bytes: 01 12 23 34 45 56 67 78 89 9A AB BC CD DE EF FF, **rellenad la siguiente tabla** 

dirección	tipo de acceso	contenido
0x986559	longword (big endian)	
0x986555	longword (little endian)	
0x986555	word (big endian)	
0x98654E	byte (little endian)	

#### Pregunta 5 (2 puntos)

**Responded** a las siguientes afirmaciones poniendo una X en el recuadro correspondiente (en la columna C si la afirmación es cierta o en la columna F si la afirmación es falsa). Cada respuesta correcta SUMA 0,2 puntos. **Cada respuesta incorrecta RESTA 0,2 puntos**. Las respuestas no contestadas no se tienen en cuenta.

С	F	Afirmación			
		Dados dos números enteros en ca2 de 4 bits 1110 y 0111, si los sumamos se produce overflow.			
		El valor de un número entero en complemento a 2 se puede calcular con la siguiente fórmula $x=-x_{n-1}\cdot 2^{n-1}+\sum_{i=0}^{n-1}x_i\cdot 2^i$			
		La siguiente instrucción genera un error de compilacion "movl %eax,2147483600(%ebx,%esi,8)"			
		En IA32 no existe ninguna instrucción con 3 operandos			
		La operación "%ebx <- 9*(%eax + 1)" se puede hacer en una sola instrucción de IA32			
		Si se modifica el registro %ch en el interior de una subrutina, NO se ha de salvar previamente en la pila			
		La instrucción "movb %ah, 6(%ebx,%ecx,8)" produce un error de compilación			
		Los structs, sin importar su tamaño, siempre se pasan por referencia cuando son parámetros de una subrutina			
		El orden de los campos de un struct NO influye en el tamaño total del mismo			
		Las instrucciones de salto modifican los flags			

curso 2007-2008 (Q1) 5/6

curso 2007-2008 (Q1) 6/6