

1 (1 punto) Un computador con palabras y direcciones de 64 bits y direccionamiento a nivel de byte, ejecuta la instrucción de una palabra:

ST .R5, [.R6]

Complete la tabla siguiente con los valores que toman en cada fase los registros indicados, y conteste a la pregunta.

	PC	R5	R6	AR	DR
Valor inicial	1000	500	2000	x	x
fetch	1008	500	2000	1000	x
fase de ejecución	1008	500	2000	2000	500

Pregunta: ¿Qué contiene la dirección de memoria 1000?

La dirección 1000 contiene La instrucción ejecutada.

2 (3 puntos) Sea un computador de palabra de 64 bits, de dos direcciones, direccionable a nivel de byte, con modelo de ejecución registro-memoria cuyos únicos modos de direccionamiento son: inmediato, directo a registro e indirecto a registro. Programe el código equivalente a las instrucciones que se indican y usando, como máximo, tres registros temporales, T1, T2 y T3:

a) ADD .R3, [--.R8], [#8[.R7--]]

b) DBNZ .R3, #24[--.R4] ; Decrementa el primer operando y si el resultado no es cero salta al segundo operando.

c) SWAP [.R1], #16[.R2++] ; Intercambia el contenido de la posición de memoria indicada por el primer operando por el de la indicada por el segundo operando.

SOLUCIÓN

a) ADD .R3, [--.R8], [#8[.R7--]]

```
SUB .R8,#8
LD .R3,[.R8]
MOV .R7,.T1
ADD .T1,#8
LD .T1,[.T1]
ADD .R3,[.T1]
SUB .R7,#8
```

b) DBNZ .R3, #24[--.R4]

```
SUB .R4,#8
MOV .R4,.T1
ADD .T1, #24
DBNZ .R3, [.T1]
```

c) SWAP [.R1], #16[.R2++]

```
MOVE .R2,.T1
ADD .T1,#16
SWAP [.R1], [.T1]
ADD .R2,#8
```

3 (6 puntos) Una tabla (LLAMADAS) contiene elementos que indican el coste de las llamadas telefónicas. Cada entrada de la tabla contiene tres enteros: el primero es el número de teléfono que ha realizado una llamada; el segundo entero contiene la tarifa que afecta a dicha llamada; el tercero la duración en minutos de la misma. La tabla no está ordenada y finaliza cuando el campo teléfono contiene el valor 0. Existe otra tabla ordenada de tarifas (TARIFAS) que contiene el precio por minuto de cada tarifa. Los identificadores de tarifas están contenidos en el rango [1-5]. Ver figuras al final del ejercicio.

Se desea generar una lista encadenada no ordenada (LISTA) en la que cada elemento contendrá tres campos: el primero es el número de teléfono; el segundo es el coste total acumulado de todas las llamadas realizadas desde ese teléfono; el tercero es un puntero que indica la dirección del siguiente elemento de la lista. Esta lista contendrá todos los números de teléfono de la primera tabla con el coste acumulado de todas las llamadas realizadas desde cada teléfono. Para generarla se supondrá que existe una subrutina de biblioteca (INSERTAR) que realiza la inserción de un elemento en la lista y que todos los parámetros se le pasan en la pila:

INSERTAR(Lista, Tlf, Coste)

- **Lista:** Es la lista en la que se desea realizar la inserción de un nuevo elemento con el teléfono y el coste que se pasan en los otros dos parámetros. Se pasa por dirección.
- **Tlf:** (entero) Es el número de teléfono que se desea insertar en la lista. Se pasa por valor.
- **Coste:**(entero) Es el coste asociado al teléfono Tlf que se desea insertar en la lista. Se pasa por valor.

Se pide realizar las siguientes subrutinas en ensamblador del MC88110:

a) (20 %) BUSCAR(Primero, Tlf) busca en la lista encadenada una entrada del teléfono Tlf. Primero es el primer elemento de la lista y se pasa por dirección (NULL si está vacía). Tlf se pasa por valor. La subrutina devuelve en r29 la dirección del elemento buscado o NULL si no se ha encontrado. No es necesario que esta subrutina implemente el marco de pila.

b) (80 %) ACTUALIZAR(Lista, Llamadas, Tarifas) genera la lista Lista a partir de las tablas Llamadas y Tarifas. Todos los parámetros se pasan en la pila por dirección. Esta rutina hará uso de las otras dos rutinas mencionadas en el enunciado.

Se supondrá que están definidas todas las macros que se han explicado en la parte teórica de la asignatura, que la subrutina llamante deja disponibles todos los registros excepto r1, r30 (SP) y r31 (FP); que la pila crece hacia direcciones de memoria decrecientes y el puntero de pila apunta a la última posición ocupada (de la misma forma que se ha utilizado en la teoría). A modo de ejemplo a continuación se muestran las tablas LLAMADAS y TARIFAS y la lista LISTA que se genera como resultado de la ejecución de la subrutina ACTUALIZAR.

Tabla LLAMADAS

Teléfono	Tarifa	Duración
609609609	2	12
601601601	2	10
609609609	1	8
609609609	3	10
0	??	??

Tabla TARIFAS

Tarifa	Precio
1	2
2	1
3	5
4	3
5	10

LISTA



SOLUCIÓN

a) La búsqueda en la lista encadenada se realizará recorriendo la lista con un puntero que se inicializará al parámetro Tlf. La búsqueda finalizará cuando se encuentra el teléfono que se pasa como parámetro en la lista o se alcanza el final de la lista (valor NULL en el puntero). A continuación se muestra el código:

```

BUSCAR:  ld r29,r30,0      ; Parámetro Primero
         ld r2,r30,4      ; Parámetro Teléfono
bucle_b: cmp r7,r29,r0    ; Si Primero = NULL se acaba
         bbl eq,r7,fin_b
         ld r3,r29,r0     ; Tlf de la lista
         cmp r7,r3,r2     ; Si Tlf de la lista = Parámetro Teléfono se acaba
         bbl eq,r7,fin_b
         ld r29,r29,8    ; Avanza al siguiente
         br bucle_b
fin_b:   jmp(r1)

```

b) Esta subrutina se resuelve recorriendo la tabla de llamadas utilizando un puntero que se almacenará como variable local en la pila. El recorrido de la tabla finalizará cuando se encuentre el valor NULL en el campo Telefono. En cada iteración se busca la entrada en la lista encadenada. Si no está se llama a INSERTAR y se almacena en el campo Coste de la lista el coste de la llamada de la tabla que se ha calculado previamente en el registro r2. Si está, se suma al campo Coste de la lista el contenido del registro r2 que es el coste de la llamada.

```

ACTUALIZAR: PUSH(r1)
            PUSH(r31)
            or r31,r30,r30
            subu r30,r30,4
bucle:     ld r20,r31,12 ; Tabla Llamadas
            ld r2,r20,r0
            cmp r7,r2,r0
            bbl eq,r7,fin
            st r20,r31,-4 ; Puntero a la tabla de llamadas
            PUSH(r2)
            ld r20,r31,8 ; Lista
            ld r20,r20,r0 ; Dir. primer elemento
            PUSH(r20)
            bsr BUSCAR
            addu r30,r30,8
            ld r20,r31,-4
            ld r2,r20,8 ; Duración
            ld r3,r20,4 ; Tarifa
            subu r3,r3,1
            mak r3,r3,0<3> ; (Tarifa-1)*8
            ld r22,r31,16 ; Tabla Tarifas
            addu r22,r22,r3 ; Tarifa buscada
            ld r4,r22,4 ; Precio
            mulu r2,r2,r4 ; Coste = Duracion*Precio
            cmp r7,r29,r0 ; Si buscar ha tenido éxito
            bbl ne,r7,cont ; salta a cont
            PUSH(r2) ; Coste
            ld r2,r20,r0 ; Tlf
            PUSH(r2)
            ld r20,r31,8 ; Lista
            PUSH(r20)
            bsr INSERTAR
            addu r30,r30,12
            ld r20,r31,-4
            addu r20,r20,12
            br bucle
cont:      ld r3,r29,4 ; Se incrementa el coste de la lista
            addu r3,r3,r2 ; con el coste de la llamada y se
            st r3,r29,4 ; almacena en el coste de la lista
            addu r20,r20,12
            br bucle
fin:       or r30,r31,r31
            POP(r31)
            POP(r1)
            jmp(r1)

```