

- 1) \* Repaso de uso de la pila Si el registro SP vale 8000h al comenzar el programa, indicar el valor del registro SP luego de ejecutar cada una de las instrucciones de la tabla, en el orden en que aparecen. Indicar, de la misma forma, los valores de los registros AX y BX.

	Instrucción	Valor del registro SP	AX	BX
1	mov ax, 5	8000h	5	?
2	mov bx, 3	8000h	5	3
3	push ax	7FFCh	5	3
4	push ax	7FFCh	5	3
5	push bx	7FFAh	5	3
6	pop bx	7FFCh	5	3
7	pop bx	7FFEh	5	5
8	pop ax	8000h	5	5

- 2) \* Llamadas a subrutinas y la pila Si el registro SP vale 8000h al comenzar el programa, indicar el valor del registro SP luego de ejecutar cada instrucción. Considerar que el programa comienza a ejecutarse con el IP en la dirección 2000h, es decir que la primera instrucción que se ejecuta es la de la línea 5 (push ax).  
Nota: Las sentencias ORG y END no son instrucciones sino indicaciones al compilador, por lo tanto no se ejecutan.

#	Instrucción	Valor del registro SP
1	org 3000h	-----
2	rutina: mov bx, 3	7FFCh
3	ret	7FFEh
4	org 2000h	-----
→ 5	push ax	7FFEh
6	call rutina	7FFCh
7	pop bx	8000h
8	hlt	8000h
9	end	-----

- 3) 1. Si el registro SP vale 8000h al comenzar el programa, indicar el contenido de la pila luego de ejecutar cada instrucción. Si el contenido es desconocido/basura, indicarlo con el símbolo "?". Considerar que el programa comienza a ejecutarse con el IP en la dirección 2000h, es decir, que la primera instrucción que se ejecuta es la de la línea 5 (call RUT). Se provee la ubicación de las instrucciones en memoria, para poder determinar la dirección de retorno de la rutina. Nota: Las sentencias ORG y END no son instrucciones, sino indicaciones al compilador, por lo tanto no se ejecutan ni tienen ubicación en memoria.

org 3000h		
RUT: mov bx, 3	Dirección 3000h	Pila: 2002h; 2006h
ret	Dirección 3002h	Pila: 2002h; 2006h
org 2000h		
call RUT	Dirección 2000h	Pila: 2002h
add cx, 5	Dirección 2002h	Pila: 2002h
call rut	Dirección 2004h	Pila: 2002h; 2006h
hlt	Dirección 2006h	Pila: 2002h; 2006h
end		

## 2. Explicar detalladamente:

### a) Las acciones que tienen lugar al ejecutarse la instrucción CALL RUT.

Al ejecutarse la instrucción CALL RUT, se guarda el valor de la posición de memoria que está en el puntero de instrucción (IP) en la pila (PUSH del IP), se asigna el valor de la posición de memoria correspondiente a la etiqueta RUT al IP y la CPU comienza a ejecutar las instrucciones de la subrutina RUT.

### b) Las acciones que tienen lugar al ejecutarse la instrucción RET.

La operación que se realiza con la instrucción ret es retornar al programa principal a partir de la instrucción siguiente a la instrucción CALL RUT. La CPU sabe a qué dirección de memoria debe retornar desde la subrutina al programa principal porque el puntero de instrucción (IP) se carga con el valor de la posición de memoria guardada en la pila (POP del IP) y, por lo tanto, la ejecución del programa sigue a partir de la instrucción siguiente a la instrucción CALL RUT.

#### 4) \* Tipos de Pasajes de Parámetros Indicar con un tilde, para los siguientes ejemplos, si el pasaje del parámetro es por registro o pila, y por valor o referencia;

	Código	Registro	Pila	Valor	Referencia
a)	mov ax, 5 call subrutina	✓		✓	
b)	mov dx, offset A call subrutina	✓			✓
c)	mov bx, 5 push bx call subrutina pop bx		/	✓	
d)	mov cx, offset A push cx call subrutina pop cx		✓		✓
e)	mov dl, 5 call subrutina	✓		✓	
f)	call subrutina mov A, dx	✓		✓	

Si tuviera que realizar el cálculo dos veces con números distintos, por ejemplo, unos guardados en variables A1, B1, C1 y otros guardados en variables A2, B2, C2, no podría reutilizar la subrutina del inciso (b) sin modificarla, aunque sí la subrutina del inciso (c).

El modo de direccionamiento de la instrucción MOV AX, [BX] es indirecto por registro y el valor que se copia en el registro AX, en este caso, es 5h.

El registro temporal denominado “ri” cumple la función de guardar, temporalmente, la dirección contenida en BX para, luego, ir a buscar el contenido de la misma.

En AX, al ejecutarse MOV AX, OFFSET RES, se guarda la dirección de la variable RES.

La variable RES a la pila se pasa por referencia y la ventaja que tiene esto (versus pasarla a la pila por valor) es poder, luego, en la subrutina SUMA, usar direccionamiento indirecto para guardar el resultado en la dirección de la variable RES.

Las instrucciones PUSH y POP trabajan para el pasaje de parámetros y para preservar el contenido de los registros.

Suma en CX el valor de AX hasta que AX llegue a 0.

El valor final de CX es 6h.

c) Dibujar las posiciones de memoria de la pila, anotando qué valores va tomando.

[illegible]

**d)** *¿Cuál será la limitación para determinar el valor más grande que se le puede pasar a la subrutina a través de AX?*

El tamaño de los registros (hasta 16 bits), si no habría un desbordamiento y los resultados serían incorrectos.