Practica 1 - Subrutinas

1 de octubre de 2024

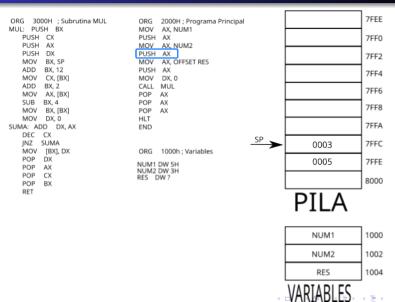
Ejercicio 7 - Multiplicación de números sin signo

- El programa utiliza una subrutina para multiplicar dos números sin signo mayores que cero.
- Se pasan los números a multiplicar por valor.
- Se para por referencia la dirección donde se debe guardar el resultado.
- Todos los parámetros se pasan a través de la pila

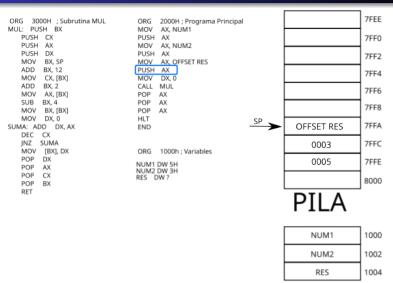
ORG 3000H ; Subrutina MUL	ORG 2000H ; Programa Principal		7FEE
MUL: PUSH BX PUSH CX PUSH AX	MOV AX, NUM1 PUSH AX		7FF0
PUSH DX PUSH DX MOV BX. SP	MOV AX, NUM2 PUSH AX MOV AX, OFFSET RES		7FF2
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX, 0		7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX		7FF6
SUB BX, 4 MOV BX, [BX]	POP AX POP AX		7FF8
MOV DX, 0 SUMA: ADD DX, AX DEC CX	HLT END		7FFA
JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables		7FFC
POP DX POP AX	NUM1 DW 5H		7FFE
POP CX POP BX	NUM2 DW 3H RES DW?		8000
RET		PILA	-
		NUM1	1000
		NUM2	1002
		RES	1004



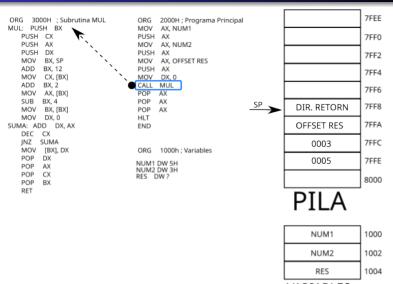
]
ORG 3000H ; Subrutina MUL	ORG 2000H; Programa Principal		7FEE
MUL: PUSH BX PUSH CX	MOV AX, NUM1 PUSH AX		7FF0
PUSH AX	MOV AX, NUM2		/ ٢٢٠
PUSH DX	PUSH AX		7FF2
MOV BX, SP	MOV AX, OFFSET RES		///2
ADD BX, 12	PUSH AX		7FF4
MOV CX, [BX]	MOV DX, 0		
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX		7FF6
SUB BX, 4	POP AX		ł
MOV BX, [BX]	POP AX		7FF8
MOV DX, 0	HLT		ł
SUMA: ADD DX, AX	END		7FFA
DEC CX			
JNZ SUMA	005 4000 - 14-1-14-		7FFC
MOV [BX], DX POP DX	ORG 1000h ; Variables	0005	1
POP AX	NUM1 DW 5H	0005	7FFE
POP CX	NUM2 DW 3H RES DW ?		1
POP BX	RES DW !		8000
RET		DILA	•
		PILA	
		1 1 - / \	
		NUM1	1000
		NU IN 42	1,000
		NUM2	1002
		RES	1004







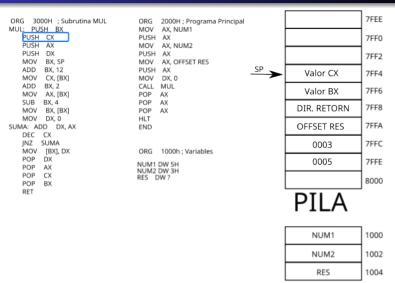




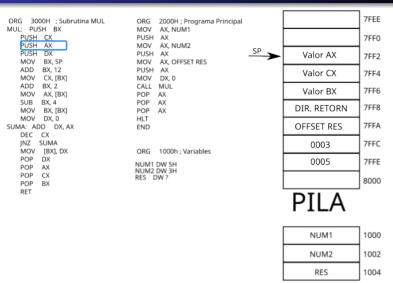


ORG 3000H ; Subrutina MUL	ORG 2000H ; Programa Principal		7FEE
MUL: PUSH BX PUSH CX	MOV AX, NUM1 PUSH AX		7FF0
PUSH AX PUSH DX MOV BX. SP	MOV AX, NUM2 PUSH AX MOV AX. OFFSET RES		7FF2
ADD BX, 12 MOV CX, [BX]	MOV AX, OFFSET RES PUSH AX MOV DX. 0		7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX	Valor BX	7FF6
SUB BX, 4 MOV BX, [BX]	POP AX POP AX	DIR. RETORN	7FF8
MOV DX, 0 SUMA: ADD DX, AX	HLT END	OFFSET RES	7FFA
DEC CX JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables	0003	7FFC
POP DX POP AX	NUM1 DW 5H	0005	7FFE
POP CX POP BX	NUM2 DW 3H RES DW ?		8000
RET		PILA	•
		LIL	
		NUM1	1000
		NUM2	1002
		RES	1004











ORG 3000H ; Subrutina MUL	ORG 2000H ; Programa Principal		7FEE
MUL: PUSH BX PUSH CX PUSH AX	MOV AX, NUM1 PUSH AX MOV AX, NUM2	Valor DX	7FF0
PUSH DX MOV BX, SP	PUSH AX MOV AX, OFFSET RES	Valor AX	7FF2
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX, 0	Valor CX	7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX	Valor BX	7FF6
SUB BX, 4 MOV BX, [BX] MOV DX, 0	POP AX POP AX HLT	DIR. RETORN	7FF8
SUMA: ADD DX, AX DEC CX	END	OFFSET RES	7FFA
JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables	0003	7FFC
POP DX POP AX	NUM1 DW 5H NUM2 DW 3H	0005	7FFE
POP CX POP BX	RES DW?		8000
RET		PILA	
		NUM1	1000
		NUM2	1002
		RES	1004

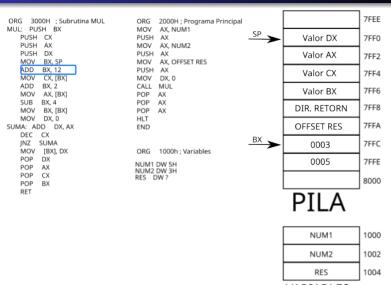
				1
ORG 3000H ; Subrutina MUL	ORG 2000H; Programa Principal			7FEE
MUL: PUSH BX PUSH CX PUSH AX	MOV AX, NUM1 PUSH AX MOV AX, NUM2	SP BX	Valor DX	7FF0
PUSH DX MOV BX, SP	PUSH AX MOV AX, OFFSET RES	DA.	Valor AX	7FF2
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX, 0		Valor CX	7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX		Valor BX	7FF6
SUB BX, 4 MOV BX, [BX] MOV DX. 0	POP AX POP AX		DIR. RETORN	7FF8
MOV DX, 0 SUMA: ADD DX, AX DEC CX	HLT END		OFFSET RES	7FFA
JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables		0003	7FFC
POP DX POP AX	NUM1 DW 5H NUM2 DW 3H		0005	7FFE
POP CX POP BX	RES DW?			8000
RET			PILA	



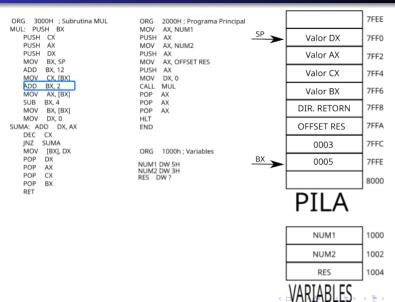
1000

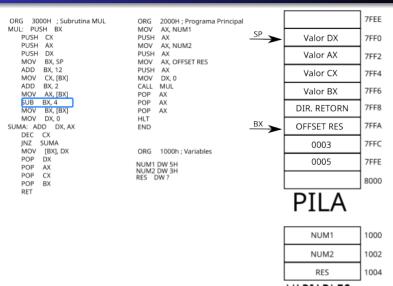
NUM1

NUM2

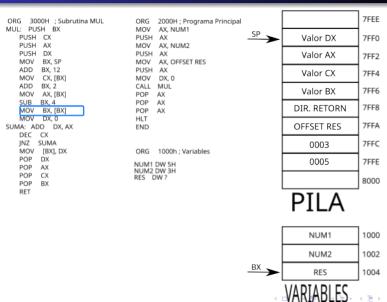


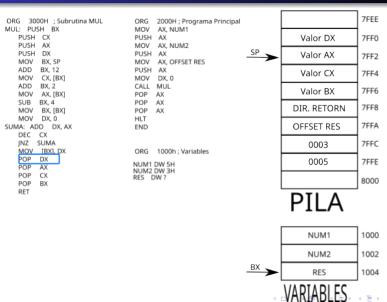


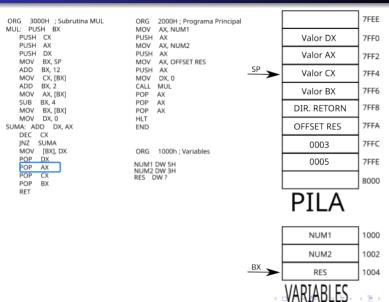




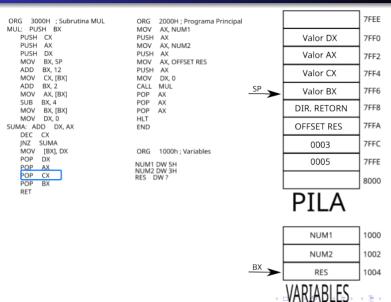






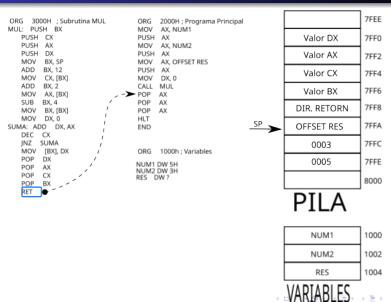






ORG 3000H ; Subrutina MUL	ORG 2000H ; Programa Principal		7FEE
MUL: PUSH BX PUSH CX PUSH AX	MOV AX, NUM1 PUSH AX MOV AX. NUM2	Valor DX	7FF0
PUSH DX MOV BX. SP	PUSH AX MOV AX, OFFSET RES	Valor AX	7FF2
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX, 0	Valor CX	7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX	Valor BX	7FF6
SUB BX, 4 MOV BX, [BX]	POP AX POP AX	DIR. RETORN	7FF8
MOV DX, 0 SUMA: ADD DX, AX DEC CX	HLT END	OFFSET RES	7FFA
JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables	0003	7FFC
POP DX	NUM1 DW 5H	0005	7FFE
POP CX	NUM2 DW 3H RES DW ?		8000
RET		PILA	•
		NUM1	1000
		NUM2	1002
		RES	1004

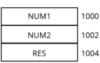




ORG 3000H ; Subrutina MUL	ORG 2000H; Programa Principal		
MUL: PUSH BX PUSH CX PUSH AX	MOV AX, NUM1 PUSH AX MOV AX, NUM2		Valor DX
PUSH DX MOV BX, SP	PUSH AX MOV AX, OFFSET RES		Valor AX
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX, 0		Valor CX
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX		Valor BX
SUB BX, 4 MOV BX, [BX]	POP AX POP AX		DIR. RETORN
MOV DX, 0 SUMA: ADD DX, AX DEC CX	HLT END		OFFSET RES
JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables	SP >	0003
POP DX POP AX	NUM1 DW 5H NUM2 DW 3H		0005
POP CX POP BX	RES DW?		
RET			PILA

7FEE 7FF0 7FF2 7FF4 7FF6 7FF8 7FFA 7FFC 7FFE 8000







ORG 3000H ; Subrutina MUL	ORG 2000H ; Programa Principal		7FEE
MUL: PUSH BX PUSH CX	MOV AX, NUM1 PUSH AX	Valor DX	7FF0
PUSH AX PUSH DX MOV BX, SP	MOV AX, NUM2 PUSH AX MOV AX, OFFSET RES	Valor AX	7FF2
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX. 0	Valor CX	7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX	Valor BX	7FF6
SUB BX, 4 MOV BX, [BX]	POP AX	DIR. RETORN	7FF8
MOV DX, 0 SUMA: ADD DX, AX	HLT END	OFFSET RES	7FFA
DEC CX JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables	0003	7FFC
POP DX POP AX	NUM1 DW 5H	0005	7FFE
POP CX POP BX	NUM2 DW 3H RES DW ?		8000
RET		PILA	•
		NUM1	1000
		NUM2	1002



1004

RES

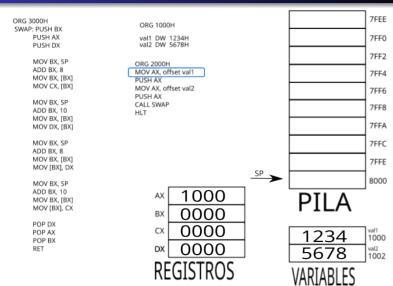
ORG 3000H ; Subrutina MUL	ORG 2000H ; Programa Principal		7FEE
MUL: PUSH BX PUSH CX	MOV AX, NUM1 PUSH AX	Valor DX	7FF0
PUSH AX PUSH DX MOV BX, SP	MOV AX, NUM2 PUSH AX MOV AX, OFFSET RES	Valor AX	7FF2
ADD BX, 12 MOV CX, [BX]	PUSH AX MOV DX, 0	Valor CX	7FF4
ADD BX, 2 MOV AX, [BX]	CALL MUL POP AX	Valor BX	7FF6
SUB BX, 4 MOV BX, [BX] MOV DX. 0	POP AX	DIR. RETORN	7FF8
MOV DX, 0 SUMA: ADD DX, AX DEC CX	HLT END	OFFSET RES	7FFA
JNZ SUMA MOV [BX], DX	ORG 1000h ; Variables	0003	7FFC
POP DX POP AX	NUM1 DW 5H	0005	7FFE
POP CX POP BX	RES DW?		8000
RET		PILA	
		NUM1	1000
		NUM2	1002
		RES	1004

Ejercicio 10 - SWAP (intercambio)

- Escribir una subrutina SWAP que intercambie dos datos de 16 bits almacenados en memoria.
- Los parámetros deben ser pasados por referencia desde el programa principal a través de la pila.

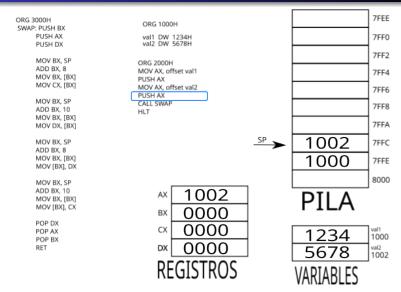
7FFF ORG 3000H ORG 1000H SWAP: PUSH BX PUSH AX val1 DW 1234H 7FF0 PUSH DX val2 DW 5678H 7FF2 MOV BX, SP ORG 2000H ADD BX. 8 MOV AX. offset val1 7FF4 MOV BX, [BX] PUSH AX MOV CX, [BX] MOV AX, offset val2 7FF6 PLISH AX MOV BX, SP CALL SWAP 7FF8 ADD BX, 10 HLT MOV BX, [BX] 7FFA MOV DX, [BX] MOV BX. SP 7FFC ADD BX. 8 MOV BX, [BX] 7FFE MOV [BX], DX SP > 8000 MOV BX, SP ADD BX, 10 0000 AX **PILA** MOV BX, [BX] MOV [BX], CX 0000 BX POP DX 0000 CX POP AX 1234 1000 POP BX 0000 RFT DX val2 5678 1002 REGISTROS

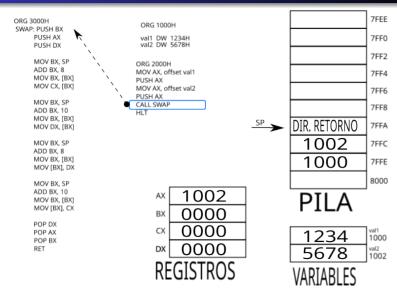
VARIABLES

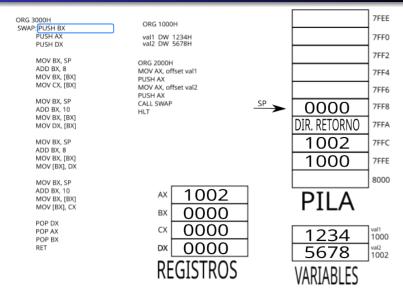


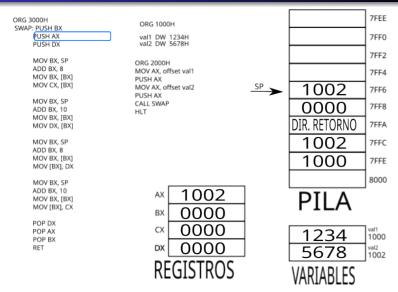
7FFF ORG 3000H ORG 1000H SWAP: PUSH BX PUSH AX val1 DW 1234H 7FF0 PUSH DX val2 DW 5678H 7FF2 MOV BX, SP ORG 2000H ADD BX. 8 MOV AX, offset val1 7FF4 MOV BX, [BX] PUSH AX MOV CX, [BX] MOV AX, offset val2 7FF6 PLISH AX MOV BX, SP CALL SWAP 7FF8 ADD BX, 10 HLT MOV BX, [BX] 7FFA MOV DX, [BX] MOV BX. SP 7FFC ADD BX. 8 1000 MOV BX, [BX] 7FFE MOV [BX], DX 8000 MOV BX, SP ADD BX, 10 1000 AX **PILA** MOV BX, [BX] MOV [BX], CX 0000 BX POP DX 0000 CX POP AX 1234 1000 POP BX 0000 RFT DX val2 5678 1002 REGISTROS VARIABLES

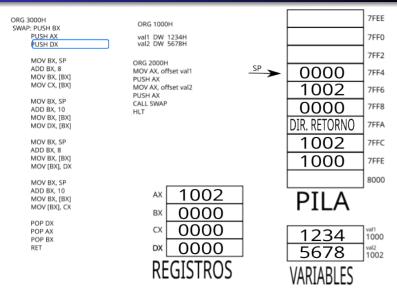
7FFF ORG 3000H ORG 1000H SWAP: PUSH BX PUSH AX val1 DW 1234H 7FF0 PUSH DX val2 DW 5678H 7FF2 MOV BX, SP ORG 2000H ADD BX. 8 MOV AX. offset val1 7FF4 MOV BX, [BX] PUSH AX MOV CX, [BX] MOV AX, offset val2 7FF6 PUSH AX MOV BX, SP CALL SWAP 7FF8 ADD BX, 10 HLT MOV BX, [BX] 7FFA MOV DX, [BX] MOV BX. SP 7FFC ADD BX. 8 1000 MOV BX, [BX] 7FFE MOV [BX], DX 8000 MOV BX, SP ADD BX, 10 1002 AX **PILA** MOV BX, [BX] MOV [BX], CX 0000 BX POP DX 0000 CX POP AX 1234 1000 POP BX 0000 RFT DX val2 5678 1002 REGISTROS VARIABLES

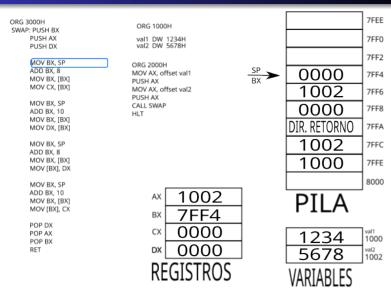


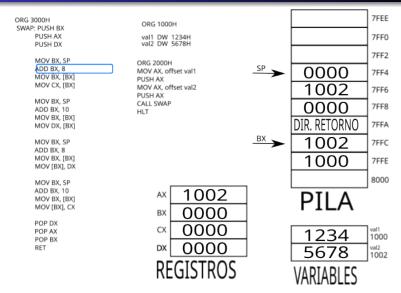


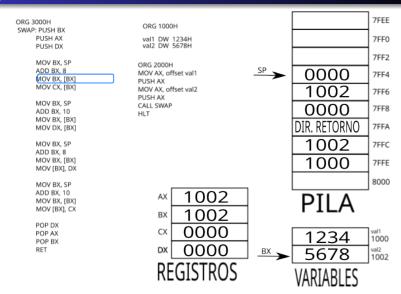


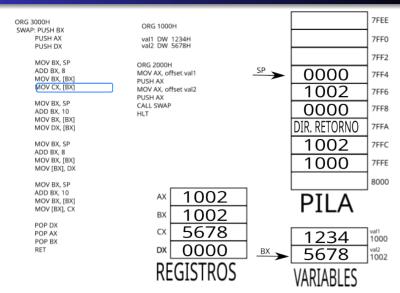


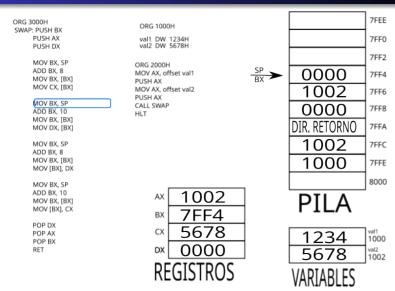


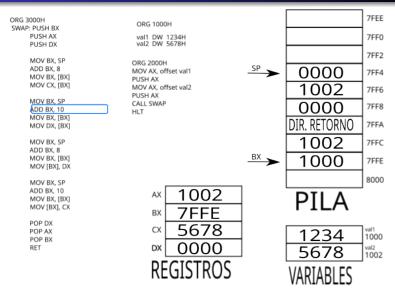


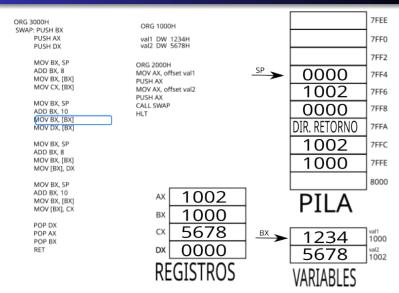


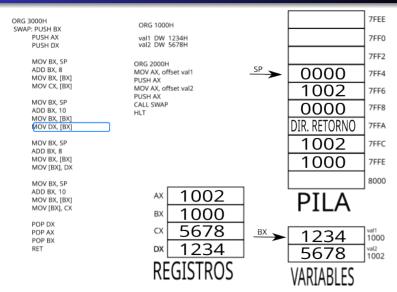


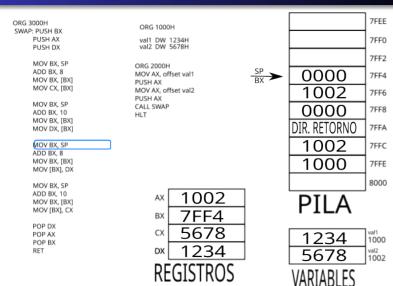


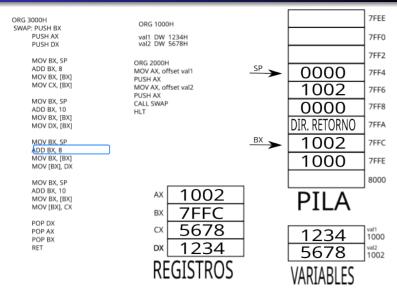


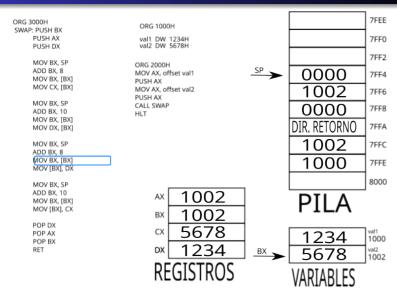


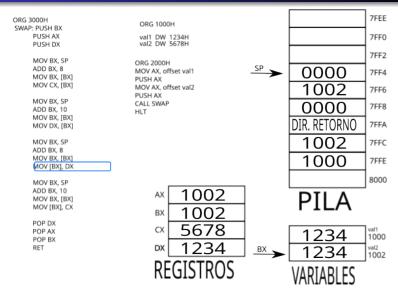




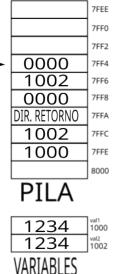


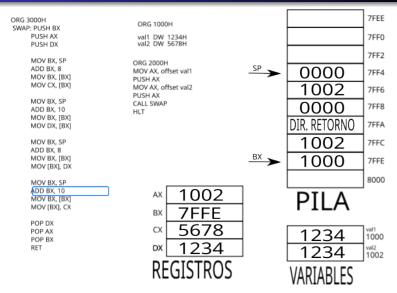


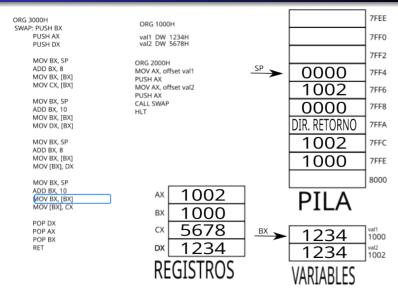


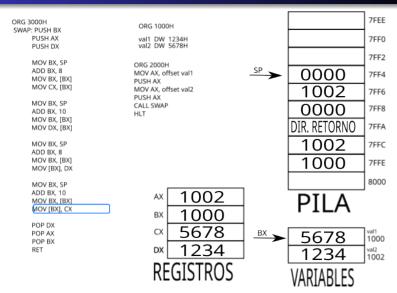


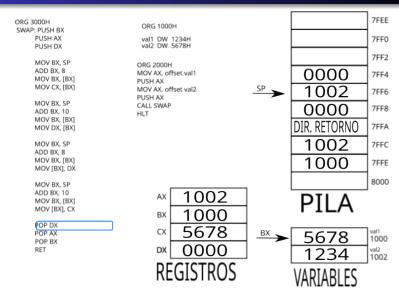
ORG 3000H ORG 1000H SWAP: PUSH BX PUSH AX val1 DW 1234H PUSH DX val2 DW 5678H MOV BX, SP ORG 2000H ADD BX. 8 MOV AX. offset val1 BX MOV BX, [BX] PUSH AX MOV CX, [BX] MOV AX, offset val2 PLISH AX MOV BX, SP CALL SWAP ADD BX, 10 HLT MOV BX, [BX] MOV DX, [BX] MOV BX. SP ADD BX. 8 MOV BX, [BX] MOV [BX], DX MOV BX, SP ADD BX, 10 1002 AX MOV BX, [BX] MOV [BX], CX 7FF4 BX POP DX 5678 CX POP AX POP BX 1234 RFT DX REGISTROS

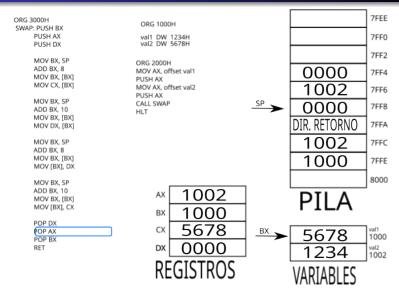




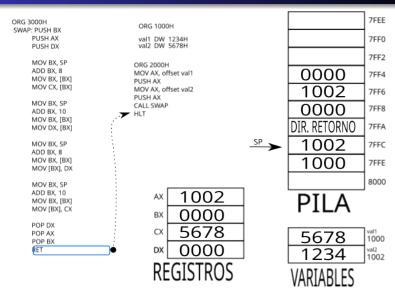








7FFF ORG 3000H ORG 1000H SWAP: PUSH BX PUSH AX val1 DW 1234H 7FF0 PUSH DX val2 DW 5678H 7FF2 MOV BX, SP ORG 2000H ADD BX. 8 0000 MOV AX. offset val1 7FF4 MOV BX, [BX] PUSH AX MOV CX, [BX] 1002 MOV AX. offset val2 7FF6 PLISH AX MOV BX, SP CALL SWAP 0000 7FF8 ADD BX, 10 HLT MOV BX, [BX] DIR. RETORNO 7FFA MOV DX, [BX] 1002 MOV BX. SP 7FFC ADD BX. 8 1000 MOV BX, [BX] 7FFE MOV [BX], DX 8000 MOV BX, SP ADD BX, 10 1002 AX PILA MOV BX, [BX] MOV [BX], CX 0000 BX POP DX 5678 CX POP AX 5678 1000 POP BX 0000 RFT DX 1234 val2 1002 REGISTROS VARIABLES



7FFF ORG 3000H ORG 1000H SWAP: PUSH BX PUSH AX val1 DW 1234H 7FF0 PUSH DX val2 DW 5678H 7FF2 MOV BX, SP ORG 2000H ADD BX. 8 0000 MOV AX. offset val1 7FF4 MOV BX, [BX] PUSH AX MOV CX, [BX] 1002 MOV AX, offset val2 7FF6 PUSH AX MOV BX, SP CALL SWAP 0000 7FF8 ADD BX, 10 HLT MOV BX, [BX] DIR. RETORNO 7FFA MOV DX, [BX] 1002 MOV BX. SP 7FFC ADD BX. 8 1000 MOV BX, [BX] 7FFE MOV [BX], DX 8000 MOV BX, SP ADD BX, 10 1002 AX PILA MOV BX, [BX] MOV [BX], CX 0000 BX POP DX 5678 CX POP AX 5678 1000 POP BX 0000 RFT DX 1234 val2 1002

REGISTROS

VARIABLES

Ejercicio 10 - SWAP (intercambio)

El programa cumple con el enunciado, pero...

- No desapila los dos parámetros de la pila
- La subrutina preserva los registros AX, BX y DX. Pero modifica los registro BX, CX y DX. No es necesario preservar AX, CX si.