Summary Exercise - Week 7

Due Nov 17 at 11:59pm **Points** 26 **Questions** 13
Available Nov 10 at 12am - Nov 17 at 11:59pm 8 days **Time Limit** 360 Minutes
Allowed Attempts 2

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	44 minutes	26 out of 26

Score for this attempt: **26** out of 26
Submitted Nov 15 at 7:08pm
This attempt took 44 minutes.

Question 1

1 / 1 pts

When passing procedure parameters on the stack, why are the following lines of code often necessary in a procedure?

```
push   ebp
mov    ebp,esp
```

○ To preserve the original EBP register value for register indirect addressing.

Correct!

● To keep additional usage of the stack within the procedure from invalidating the stack offsets.

○ Because the procedure might change the EBP register value.

○ They are never necessary.

Question 2

1 / 1 pts

When passing parameters to a procedure on the stack, it is usually okay to change the value of the EBP register within the procedure.

○ True

Correct! ● False

Question 3

1 / 1 pts

The following two instructions are equivalent.

```
ret
ret 4
```

○ True

Correct! ● False

Question 4

1 / 1 pts

The RET instruction pops the top of the stack into what register?

Correct! ● EIP

○ EBP

○ ESP

○ It does not pop the top of the stack into a register.

Question 5

1 / 1 pts

Given *list*, an array of WORDs, what element is addressed by *list[8]*?

Hint: It's Love.

○ 4th Element

○ 9th Element

○ 8th Element

Correct! ● 5th Element

Question 6

1 / 1 pts

Arrays are stored in ____________ memory.

○ Random

Correct! ● Contiguous

○ Disjoint

Question 7

1 / 1 pts

Given the following register states, and using Register Indirect Addressing, which of the following lines of code will move the 11th element of the *list* array (of DWORDs) to the EAX register?

EDX register contans the address of the first element of *list*.
ESI register contains the address of the eleventh element of *list*.
EBX register contains the value 40,

○ mov eax, list[esi]

○ mov eax, [edx + ebx]

Correct! ● mov eax, [esi]

○ mov eax, list[ebx]

Question 8

1 / 1 pts

Given the following register states, and using Base Indexed Addressing, which of the following lines of code will move the 11th element of the *list* array (of DWORDs) to the EAX register?

EDX register contans the address of the first element of *list*.
ESI register contains the address of the eleventh element of *list*.
EBX register contains the value 40,

○ mov eax, list[ebx]

Correct! ● mov eax, [edx + ebx]

○ mov eax, [esi]

○ mov eax, list[esi]

Question 9

4 / 4 pts

The following instruction will increment the stack pointer (ESP) by how many bytes? (Ignore the .0 after the number. Canvas insists on pushing decimals even when kindly asked not to).

```
ret 13
```

Correct!

17

orrect Answer

17

Question 10

4 / 4 pts

Suppose that you are given the following program (with memory addresses shown on the left).
What hexadecimal value does EIP hold immediately after "inc EAX" has executed?

```
.data
0x100   x   DWORD  153461
0x104   y   WORD   37
0x105   z   WORD   90



.code
main PROC
0x12   push  x
0x17   mov   AX, y
0x1C    shl   AX, 16
0x1C   mov   AX, z
0x21   call  someProcedure
0x26   inc   EAX
0x2B   mov   EBX, z
0x30   xor   EAX, EBX
0x35   exit
main ENDP
END MAIN
```

Correct!

0x2B

orrect Answers

x2B

0x2B

2Bh

2B

Question 11

4 / 4 pts

Suppose that you are given the following program.
Inside *someProcedure*, what numerical operand should be used with the *RET* instruction?

```
.data
x    DWORD   153461
y    WORD    37
z    WORD    90



.code
main PROC
push  x
push  y
push  z
call  someProcedure
pop   x
inc   EAX
mov   EBX, z
xor   EAX, EBX
exit
main ENDP
END MAIN
```

Correct!

4

orrect Answers

4

Question 12

3 / 3 pts

For this problem, suppose that you are working with the partial data segment given below. Assume that the memory address of **balance** is 0x44. What hexadecimal address belongs to the **first** item in **history**?

```
HISTLIMIT = 100

.data
balance    DWORD   0
account    WORD    ?
history    WORD    HISTLIMIT DUP(?)
isValid    BYTE    0
```

Correct!

0x4A

orrect Answers

4Ah

x4A

4A

0x4A

Question 13

3 / 3 pts

For this problem, suppose that you are working with the partial data segment given below. Assume that the memory address of **balance** is 0x44. What hexadecimal address belongs to the **last** item in **history**?

```
HISTLIMIT = 100

.data
balance    DWORD   0
account    WORD    ?
history    WORD    HISTLIMIT DUP(?)
isValid    BYTE    0
```

Correct!

0x110

orrect Answers

0x0110

x0110

0x110

0110

110

0110h

110h

x110

Quiz Score: **26** out of 26