

Question 1	3 / 3 pts
<pre>.DATA n BYTE 15h .CODE main PROC MOV BL, 9 MOV BH, n AND BL, BH INVOKE ExitProcess, 0 main ENDP END main</pre> <p>1. What is the value of BL after the program executes?</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 8 <input type="radio"/> Not enough information to answer <input type="radio"/> None of the above</p>	

Question 2	3 / 3 pts
<pre>.DATA n BYTE 15h .CODE main PROC MOV BL, 9 MOV BH, n AND BL, BH INVOKE ExitProcess, 0 main ENDP END main</pre> <p>2. What is the value of BH after the program executes?</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 8 <input type="radio"/> Not enough information to answer <input checked="" type="radio"/> None of the above</p>	

Question 3	3 / 3 pts
<pre>.DATA n BYTE 15h .CODE main PROC MOV BL, 9 MOV BH, n AND BL, BH INVOKE ExitProcess, 0 main ENDP END main</pre> <p>3. Replace the line "n BYTE 15h" with "n BTYE 5h" What is the value of BL after the program executes?</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 8 <input type="radio"/> Not enough information to answer <input type="radio"/> None of the above</p>	

Question 4	3 / 3 pts
<pre>.DATA array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100 v SBYTE 3 .CODE main PROC CLD MOV EBX, 1 MOV AL, v MOV ECX, LENGTHOF array MOV EDI, OFFSET array REPNE SCASB CMP ECX, 0 JNZ D MOV EBX, 0 D: INVOKE ExitProcess, 0 main ENDP END main</pre>	

END main	
<p>Note:</p> <p>The SCAS instruction is used to scan a string (SCAS = SScan A String). It compares the content of the accumulator(AL, AX, or EAX) against the current value pointed at by ES:[EDI]</p> <p>When used together with the REPNE prefix (REPeat while NOT Equal), SCAS scans the string searching for the first string element which is equal to the value in the accumulator. The instruction SCASB searches the memory for the byte in EAX, starting EDI:</p> <p>Compare a byte of memory with the AL register.</p> <p>Increments EDI by 1 after each byte comparison</p> <p>4. What is the value of ECX after the program executes?</p> <p><input type="radio"/> 0 <input type="radio"/> 2 <input checked="" type="radio"/> 4 <input type="radio"/> 7 <input type="radio"/> 0A</p>	

Question 5

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
v SBYTE 3
.CODE
main PROC
CLD
MOV EBX, 1
MOV AL, v
MOV ECX, LENGTHOF array
MOV EDI, OFFSET array
REPNE SCASB
CMP ECX, 0
JNZ D
MOV EBX, 0
D:
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

The SCAS instruction is used to scan a string (SCAS = SScan A String). It compares the content of the accumulator(AL, AX, or EAX) against the current value pointed at by ES:[EDI]

When used together with the REPNE prefix (REPeat while Not Equal), SCAS scans the string searching for the first string element which is equal to the value in the accumulator. The instruction SCASB searches the memory for the byte in EAX, starting EDI:

Compare a byte of memory with the AL register.

Increments EDI by 1 after each byte comparison

5. What is the value of EBX after the program executes?

- 0
- 2
- 4
- Not enough information to answer
- None of the above

Question 6

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
v SBYTE 3
.CODE
main PROC
CLD
MOV EBX, 1
MOV AL, v
MOV ECX, LENGTHOF array
MOV EDI, OFFSET array
REPNE SCASB
CMP ECX, 0
JNZ D
MOV EBX, 0
D:
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

The SCAS instruction is used to scan a string (SCAS = SScan A String). It compares the content of the accumulator(AL, AX, or EAX) against the current value pointed at by ES:[EDI]

When used together with the REPNE prefix (REPeat while Not Equal), SCAS scans the string searching for the first string element which is equal to the value in the accumulator. The instruction SCASB searches the memory for the byte in EAX, starting EDI:

Compare a byte of memory with the AL register.

Increments EDI by 1 after each byte comparison

6. What is the value of EDI after the program executes?

- Increments EDI by 1 after each byte comparison
6. What is the value of EDI after the program executes?
- 2
 - 4
 - 5
 - Not enough information to answer
 - None of the above

Question 7

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
v SBYTE 3
.CODE
main PROC
CLD
MOV EBX, 1
MOV AL, v
MOV ECX, LENGTHOF array
MOV EDI, OFFSET array
REPNE SCASB
CMP ECX, 0
JNZ D
MOV EBX, 0
D:
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

The SCAS instruction is used to scan a string (SCAS = SScan A String). It compares the content of the accumulator(AL, AX, or EAX) against the current value pointed at by ES:[EDI]

When used together with the REPNE prefix (REPeat while Not Equal), SCAS scans the string searching for the first string element which is equal to the value in the accumulator. The instruction SCASB searches the memory for the byte in EAX, starting EDI:

Compare a byte of memory with the AL register.

Increments EDI by 1 after each byte comparison

7. What is the value of AL after the program executes?

- Increments EDI by 1 after each byte comparison
7. What is the value of AL after the program executes?
- 0
 - 2
 - 3
 - 1
 - None of the above

Question 8

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
v SBYTE 3
.CODE
main PROC
CLD
MOV EBX, 1
MOV AL, v
MOV ECX, LENGTHOF array
MOV EDI, OFFSET array
REPNE SCASB
CMP ECX, 0
JNZ D
MOV EBX, 0
D:
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

The SCAS instruction is used to scan a string (SCAS = SScan A String). It compares the content of the accumulator(AL, AX, or EAX) against the current value pointed at by ES:[EDI].

When used together with the REPNE prefix (REPeat while Not Equal), SCAS scans the string searching for the first string element which is equal to the value in the accumulator. The instruction SCASB searches the memory for the byte in EAX, starting EDI:

Compare a byte of memory with the AL register.

Increments EDI by 1 after each byte comparison

Compare a byte of memory with the AL register.

Increments EDI by 1 after each byte comparison

8. Replace the line "v SBYTE 3" with "v SBYTE 2". What is the value of ECX after the program executes?

- 0
- 2
- 4
- 0A
- None of the above

Question 11

3 / 3 pts

```
.DATA
n BYTE 15h
.CODE
main PROC
MOV BL, 9
MOV BH, n
XOR BL, BH
INVOKE ExitProcess, 0
main ENDP
END main
```

11. What is the value of BH after the program executes?

- 9
- 15
- 1C
- 0C
- None of the above

Question 9

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
v SBYTE 3
.CODE
main PROC
CLD
MOV EBX, 1
MOV AL, v
MOV ECX, LENGTHOF array
MOV EDI, OFFSET array
REPNE SCASB
CMP ECX, 0
JNZ D
MOV EBX, 0
D:
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

The SCAS instruction is used to scan a string (SCAS = SScan A String). It compares the content of the accumulator(AL, AX, or EAX) against the current value pointed at by ES:[EDI].

When used together with the REPNE prefix (REPeat while Not Equal), SCAS scans the string searching for the first string element which is equal to the value in the accumulator. The instruction SCASB searches the memory for the byte in EAX, starting EDI:

Compare a byte of memory with the AL register.

Increments EDI by 1 after each byte comparison

Increments EDI by 1 after each byte comparison

9. Replace the line "v SBYTE 3" with "v SBYTE 2". What is the value of EBX after the program executes?

- 0
- 2
- 4
- 1
- None of the above

Question 10

3 / 3 pts

```
.DATA
n BYTE 15h
.CODE
main PROC
MOV BL, 9
MOV BH, n
XOR BL, BH
INVOKE ExitProcess, 0
main ENDP
END main
```

10. What is the value of BL after the program executes?

- 9
- 15
- 1C
- Not enough information to answer
- None of the above

Question 13

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ECX, LENGTHOF array
MOV EAX, 0
L:
DEC ECX
CMP array[ECX], 0
JL A
INC AH
JMP B
A:
INC AL
B:
CMP ECX, 0
JNZ L
INVOKE ExitProcess, 0
main ENDP
END main
```

13. What is the value of ECX after the program executes?

- 2
- 1
- 0
- Not enough information to answer
- None of the above

Question 11

3 / 3 pts

```
.DATA
n BYTE 15h
.CODE
main PROC
MOV BL, 9
MOV BH, n
XOR BL, BH
INVOKE ExitProcess, 0
main ENDP
END main
```

11. What is the value of BH after the program executes?

- 9
- 15
- 1C
- 0C
- None of the above

Question 12

3 / 3 pts

```
.DATA
n BYTE 15h
.CODE
main PROC
MOV BL, 9
MOV BH, n
XOR BL, BH
INVOKE ExitProcess, 0
main ENDP
END main
```

12. Replace the line "n BYTE 15h" with "n BYTE 5h". What is the value of BL after the program executes?

- 9
- 15
- 1C
- 0C
- None of the above

Question 14

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ECX, LENGTHOF array
MOV EAX, 0
L:
DEC ECX
CMP array[ECX], 0
JL A
INC AH
JMP B
A:
INC AL
B:
CMP ECX, 0
JNZ L
Invoke ExitProcess, 0
main ENDP
END main
```

14. What is the value of AL after the program executes?

- 3
- 5
- 0A
- 19
- None of the above

Question 17

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ECX, LENGTHOF array
MOV EAX, 0
L:
DEC ECX
CMP array[ECX], 0
JL A
INC AH
JMP B
A:
INC AL
B:
CMP ECX, 0
JNZ L
Invoke ExitProcess, 0
main ENDP
END main
```

17. Replace the line "array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100" with "array SBYTE 8, 24, 53, 100".

```
JNZ L
Invoke ExitProcess, 0
main ENDP
END main
```

17. Replace the line "array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100" with "array SBYTE 8, 24, 53, 100".

After the program executes what will the value of AL be?

- 2
- 4
- 8
- 24
- None of the above

Question 15

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ECX, LENGTHOF array
MOV EAX, 0
L:
DEC ECX
CMP array[ECX], 0
JL A
INC AH
JMP B
A:
INC AL
B:
CMP ECX, 0
JNZ L
Invoke ExitProcess, 0
main ENDP
END main
```

15. What is the value of AH after the program executes?

- 3
- 5
- 0A
- 19
- None of the above

Question 16

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ECX, LENGTHOF array
MOV EAX, 0
L:
DEC ECX
CMP array[ECX], 0
JL A
INC AH
JMP B
A:
INC AL
B:
CMP ECX, 0
JNZ L
Invoke ExitProcess, 0
main ENDP
END main
```

16. Replace the line "array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100" with "array SBYTE 8, 24, 53, 100". After the program executes what will the value of AH be?

- 2
- 4
- 8
- 24

Question 18

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ESI, 0
MOV EDX, 0
MOV ECX, 2
MOV EAX, LENGTHOF array
DIV ECX
MOV EDI, EAX
L:
MOV DL, array[ESI]
XCHG DL, array[EDI]
XCHG DL, array[ESI]
INC ESI
INC EDI
SUB EAX, 1
JNZ L
Invoke ExitProcess, 0
main ENDP
END main
```

Note:

Dividend	Divisor	Quotient	Reminder
AX r/m8	AL	AH	
DX:AX r/m16	AX	DX	
EDX:EAX /m32	EAX	EDX	

AX

DX:AX

18. What is the value of EAX after the program executes?

- 8
- 4
- 2
- 0
- None of the above

Question 19

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ESI, 0
MOV EDX, 0
MOV ECX, 2
MOV EAX, LENGTHOF array
DIV ECX
MOV EDI, EAX
L:
MOV DL, array[ESI]
XCHG DL, array[EDI]
XCHG DL, array[ESI]
INC ESI
INC EDI
SUB EAX, 1
JNZ L
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

Dividend | Divisor | Quotient | Reminder

AX	r/m8	AL	AH
DX:AX	r/m16	AX	DX
EDX:EAX	/m32	EAX	EDX

Note:

Dividend | Divisor | Quotient | Reminder

AX	r/m8	AL	AH
DX:AX	r/m16	AX	DX
EDX:EAX	/m32	EAX	EDX

19. What is the value of ESI after the program executes?

- 0A
 8
 5
 Not enough information to answer
 None of the above

Question 20

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ESI, 0
MOV EDX, 0
MOV ECX, 2
MOV EAX, LENGTHOF array
DIV ECX
MOV EDI, EAX
L:
MOV DL, array[ESI]
XCHG DL, array[EDI]
XCHG DL, array[ESI]
INC ESI
INC EDI
SUB EAX, 1
JNZ L
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

Dividend | Divisor | Quotient | Reminder

AX	r/m8	AL	AH
DX:AX	r/m16	AX	DX
EDX:EAX	/m32	EAX	EDX

Note:

Dividend | Divisor | Quotient | Reminder

AX	r/m8	AL	AH
DX:AX	r/m16	AX	DX
EDX:EAX	/m32	EAX	EDX

20. What is the value of EDI after the program executes?

- 0A
 8
 5
 Not enough information to answer
 None of the above

Question 21

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ESI, 0
MOV EDX, 0
MOV ECX, 2
MOV EAX, LENGTHOF array
DIV ECX
MOV EDI, EAX
L:
MOV DL, array[ESI]
XCHG DL, array[EDI]
XCHG DL, array[ESI]
INC ESI
INC EDI
SUB EAX, 1
JNZ L
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

Dividend | Divisor | Quotient | Reminder

AX	r/m8	AL	AH
DX:AX	r/m16	AX	DX
EDX:EAX	/m32	EAX	EDX

Note:

Dividend | Divisor | Quotient | Reminder

AX	r/m8	AL	AH
DX:AX	r/m16	AX	DX
EDX:EAX	/m32	EAX	EDX

21. What is the value of ECX after the program executes?

- 8
 4
 2
 0
 None of the above

Question 22

3 / 3 pts

```
.DATA
array SBYTE -19, -3, -1, 0, 1, 3, 8, 24, 53, 100
.CODE
main PROC
MOV ESI, 0
MOV EDX, 0
MOV ECX, 2
MOV EAX, LENGTHOF array
DIV ECX
MOV EDI, EAX
L:
MOV DL, array[ESI]
XCHG DL, array[EDI]
XCHG DL, array[ESI]
INC ESI
INC EDI
SUB EAX, 1
JNZ L
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

Dividend | Divisor | Quotient | Reminder

AX r/m8 AL AH

DX:AX r/m16 AX DX

EDX:EAX /m32 EAX EDX

Question 23

3 / 3 pts

```
Rectangle STRUCT
len WORD 2
wid WORD 2
Rectangle ENDS
.DATA
r Rectangle <> ;Replace here
.CODE
main PROC
MOV AX, r.len
MUL r.wid
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:

Multiplicand | Multiplier | Product

AL r/m8 AX

AX r/m16 DX:AX

EAX r/m32 EDX:EAX

Note:

Multiplicand | Multiplier | Product

AL r/m8 AX

AX r/m16 DX:AX

EAX r/m32 EDX:EAX

23. What is the value of AX after the program executes?

 0 2 4 Not enough information to answer None of the above**Question 24**

3 / 3 pts

```
Rectangle STRUCT
len WORD 2
wid WORD 2
Rectangle ENDS
.DATA
r Rectangle <> ;Replace here
.CODE
main PROC
MOV AX, r.len
MUL r.wid
INVOKE ExitProcess, 0
main ENDP
END main
```

Note:
Multiplicand | Multiplier | Product
AL r/m8 AX
AX r/m16 DX:AX
EAX r/m32 EDX:EAX

Note:

Multiplicand | Multiplier | Product

AL r/m8 AX

AX r/m16 DX:AX

EAX r/m32 EDX:EAX

24. What is the value of DX after the program executes?

 0 2 4 Not enough information to answer None of the above

Question 25

3 / 3 pts

```
Rectangle STRUCT
    len WORD 2
    wid WORD 2
Rectangle ENDS
.DATA
    r Rectangle <> ;Replace here
.CODE
    main PROC
        MOV AX, r.len
        MUL r.wid
        INVOKE ExitProcess, 0
    main ENDP
END main
```

Note:

Multiplicand Multiplier Product
AL r/m8 AX
AX r/m16 DX:AX
EAX r/m32 EDX:EAX

Note:

Multiplicand Multiplier Product
AL r/m8 AX
AX r/m16 DX:AX
EAX r/m32 EDX:EAX

25. Replace the line "r Rectangle <>" with "r Rectangle <1,>". What is the value of AX after the program executes?

- 0
- 2
- 4
- Runtime Error
- None of the above

Question 26

3 / 3 pts

```
Rectangle STRUCT
    len WORD 2
    wid WORD 2
Rectangle ENDS
.DATA
    r Rectangle <> ;Replace here
.CODE
    main PROC
        MOV AX, r.len
        MUL r.wid
        INVOKE ExitProcess, 0
    main ENDP
END main
```

Note:

Multiplicand Multiplier Product
AL r/m8 AX
AX r/m16 DX:AX
EAX r/m32 EDX:EAX

Note:

Multiplicand Multiplier Product
AL r/m8 AX
AX r/m16 DX:AX
EAX r/m32 EDX:EAX

26. Replace the line, "r Rectangle <>" with "r Rectangle <1,>". What is the value of AX after the program executes?

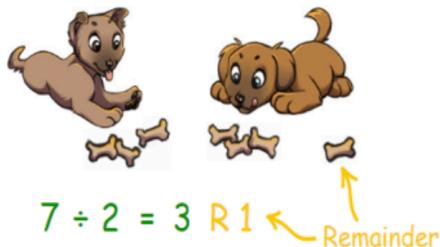
- 0
- 2
- 4
- Runtime Error
- None of the above

Question 27

22 / 22 pts

Section II. Code writing

27. In Math, a division can be denoted as: dividend ÷ divisor = quotient.... remainder. For example



Please implement a division: $123 \div 10$ (they are decimal numbers). Your program will not be terminated until quotient is less than divisor. Please push the remainder separately into stack. For example,

1. $123 \div 10 = 12$ (quotient)....3(remainder) push 3 into stack
2. $12 \div 10 = 1$ (quotient)....2(remainder) push 2 into stack
3. 1 is less than 10, your program ends and also push 1 into stack

In your program, please use (1) AL to save dividend; (2) DL to save divisor; (3) BX to save remainder. Assume (1) all registers are already initialized with 0; and (2) decimal numbers can be stored in registers.

main PROC

?

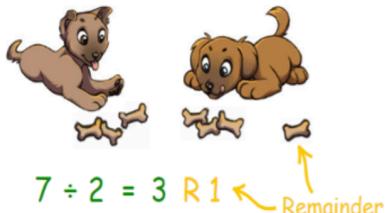
...

main ENDP**END main****Question 27**

22 / 22 pts

Section II. Code writing

27. In Math, a division can be denoted as: dividend ÷ divisor = quotient.... remainder. For example



Please implement a division: $123 \div 10$ (they are decimal numbers). Your program will not be terminated until quotient is less than divisor. Please push the remainder separately into stack. For example,

1. $123 \div 10 = 12$ (quotient)....3(remainder) push 3 into stack
2. $12 \div 10 = 1$ (quotient)....2(remainder) push 2 into stack
3. 1 is less than 10, your program ends and also push 1 into stack

In your program, please use (1) AL to save dividend; (2) DL to save divisor; (3) BX to save remainder. Assume (1) all registers are already initialized with 0; and (2) decimal numbers can be stored in registers.

main PROC

?

...

main ENDP**END main****Write down code to finish the part between the main PROC and ENDP****Your Answer:**

```
mov al, 123  
mov dl, 10  
S:  
    cmp al, dl  
    jl R  
    sub al, dl  
    inc ah  
    jmp S
```

```
R:  
    mov bl, al  
    push bx  
    mov ebx 0  
  
    cmp ah, 0  
    je T  
  
    mov al, ah  
    mov ah, 0  
    jmp S
```

T:

```
R:  
    mov bl, al  
    push bx  
    mov ebx 0  
  
    cmp ah, 0  
    je T  
  
    mov al, ah  
    mov ah, 0  
    jmp S
```

T:**Write down code to finish the part between the main PROC and ENDP**