

This exam in both sides, make sure to answer all the questions. "No Calculators allowed"

1. (10 Points) Indicate whether the following statements are syntactically correct or incorrect, and If not correct explain why or give a correction :

- | | |
|-----------------------|-----------------------|
| a. add ebx,ax | f. sub bl,22 |
| b. var1 word 5 | g. movzx ax, bx |
| c. inc word ptr [esi] | h. movsx ax, al |
| d. mov 0ffh, cx | i. var word 3 dup (6) |
| e. pop 01h | j. .code? |

2. (10 Points) Use the following data for the next questions:

```
.data
val1 BYTE 05h
val2 WORD 1234h
val3 DWORD 0FFFFh
val4 WORD 7F7Fh
```

- (1 point) Write an instruction that increments val2.
- (1 point) Write an instruction that subtracts val3 from EAX.
- (2 point) Write instructions that subtract val4 from val2.
- (1 point) If val2 is incremented by 1 using the ADD instruction, what will be the values of the Carry and Sign flags?
- (1 point) If val4 is incremented by 1 using the ADD instruction, what will be the values of the Overflow and Sign flags?
- (4 points) Consider AX=07ffh, write down the values of the Carry, Sign, Zero, and Overflow flags after each instruction has executed: (*Instructions are separated not dependent on each other*)

f.1. add al,1h	;	CF =	SF =	ZF =	OF =
f.2. inc al	;	CF =	SF =	ZF =	OF =
f.3. sub ah,08	;	CF =	SF =	ZF =	OF =
f.4. neg ax	;	CF =	SF =	ZF =	OF =

3. (10 Points) Complete the following sentences:

- The 8-bit signed binary form of the decimal "-13" is
- The Hexadecimal form of the binary number "1010 0111" is
- The Real-address mode can address a range of, while the primary Protected-address mode can address a range of
- is a utility program that combines individual files into a single executable program.
- In Assembly programs, the directive identify the data section, while the identify the code section.
- The is an informative file optionally generated by assembling an assembly program and includes information such as Source code, Machine code, Segments offset.
- The regular steps required to transform an Assembly program to ".exe" file are, Assembling,, and

4. (10 Points) Trace the following Assembly program then answer the questions:

```
1.  INCLUDE Irvine32.inc
2.
3.  .data
4.  byteVal      byte 01h,02h,03h
5.  wordVal      word 1122h,3344h
6.  dwordVal     dword 00AABBCCCh
7.  ptr          dword dwordVal
8.
9.  .code
10. main PROC
11.
12.         mov eax,0
13.         mov ebx,0
14.
15.         mov esi, offset byteVal
16.         inc byte ptr [esi]
17.         inc esi
18.
19.         mov ah, byte ptr wordVal
20.         mov bx, word ptr byteVal+2
21.         mov cx, lengthof wordVal
22.         mov dx, word ptr ptr;
23.
24.         call dumpregs
25.
26.         exit
27. main ENDP
    END main
```

- What is the value of the registers AH, BX, CX, and DX after executing this program.
- What is the value stored in `[ESI]` after executing this program
- What is the value of CX if we change line 21 into “mov cx, sizeof wordVal”
- Write a piece of Assembly code (not a complete program) to dump the contents of memory for the “.data” section.
- Write the contents of memory in bytes for “.data” section after executing the program.
Hint: this question asks about the proper bytes order for the stored variables in memory

5. (5 Points) Using `Irvine32.lib`, write a complete assembly language program to prompt for and input the temperature in degrees Fahrenheit, calculate the degrees in Celsius, and then output the degrees in Celsius. The equation to be used is $C = (F-32)/9*5$, where C stands for Celsius and F stands for Fahrenheit. The form of the input and output can be found below.

```
Enter the degrees in Fahrenheit: 100
The degrees in Celsius is: 354
```

6. (5 points) Using `Irvine32.lib`, write a complete assembly language program to display the odd numbers from [0-100], and leave 3 spaces after each number. The form of the output should be as following:

```
1 3 5 7 ..... 99
```