Question1

what will be the values of register CL and the specified flags after executing the following instructions:
mov CL,40h ;   assume the values are signed integer
add CL,40h ;   assume the values are signed integer

CL=, SF=, ZF=, OF=, CF=
Note: 0 means a flag is clear and 1 means it is set.

Answer:
Answer:
CL = 80h
SF = 1 , the result 80 in binary 1000 0000, Most significant bit is 1, so Sign flag is 1
ZF= 0 , because the result is NOT zero. Zero flag is 1 when result is Zero.
PF= 0, the result 80 in binary 1000 0000, count the number of 1's . the count is 1.
When count is even PF = 1, when count is ODD PF = 0
CF = 0, the result doesn't have any carry.

Question 2

The data segment of a assembly program is as follows:

.data

aVal DWORD 12

bVal DWORD 13

and eax register contains 4, ebx register contains 5, and ecx register contains 3.

Write assembly code to evaluate the following expression.

aVal = (bVal-ecx) + (eax - ebx)

Answer: MOV, ADD and SUB cannot take two variables, at least one register needed.

```
MOV EAX, 4
MOV EBX, 5
MOV ECX, 3

SUB bVal, ECX
SUB EAX, EBX
ADD EAX, bVal
MOV aVal, EAX
```

Question 3:

Consider the following code:

```
.386
.model flat, stdcall .stack 4096

ExitProcess PROTO, dwExitCode : DWORD

.data
aVal SDWORD -6

bVal SWORD 19h

cVal DWORD 17h

.code
mov edx, aVal
add edx, edx
mov eax, 0FFFFFFFFh

mov ax, bVal
sub edx, eax
```

Show the content of edx and eax after executing each instruction in Hexadecimal.

Answer:

Step 1:

2’s complement of -6 (32 bit)

6 in Hex: 0 0 0 0 0 0 0 6 (32 bit)

```
1’st Complement:    _F F F F  F F F F
-                    0 0 0 0 0 0 0 6
                     F F F F  F F F_9
2’s complement:    F F F F  F F F_9
+                                 1
                   F F F F  F F F_A
```

After **mov edx, aVal**

EDX: F F F F F F F A

EAX: no change

Step 2:

add edx, edx

```
carry             1  1 1 1 1 1 1 1

                     F F F F  F F F A
+                    F F F F  F F F A
--------------------------------------
                  1  F F F F  F F  F 4
```

The first one discard.

So

EDX: F F F F F F F 4

EAX: no change

Step 3

mov eax, 0FFFFFFFFh

EDX: F F F F F F F 4

EAX: **0FFFFFFFF**

Step 4

mov ax, bVal (bVal SWORD 19h)

EDX: F F F F F F F 4

EAX: **FFFF 0019**

Step 5

sub edx, eax

EAX: **FFFF 0019(no change)**

First calculate 2's complement of EAX (because of sub)

1’s complement of EAX:

```
        FFFFFFFF
    -   FFFF0019
        0000FFE6
```
2’s complement of EAX
```
        0000FFE6
+              1
        0000FFE7
```

sub edx, eax

EDX + 2’s complement of EAX

```
Carry       1 1 1 1 1 1 1 1
              F F F F  F F F 4
        +    0 0 0 0 F F E  7
            1 0 0 0 0 F F D B
```
The first one discard

So the EDX: 0 0 0 0 F F D B

question 4.

Read the following assembly instructions carefully.

```
mov eax, 10h
mov ebx, 11h
add ecx, eax

add ecx, ebx
```

What is content of the ecx register after you execute these four instructions. Justify your answer.

Answer:

eax contains 10h,

ebx contains 11h.

We add content of eax with content of ecx.

The ecx register contains unknown value (garbage value).

So, the result can be anything. (depend on the initial value of ECX)

Question 5.

A computer program has three types of instructions. 30% instructions of the program are load instructions, 20% instructions of the program are arithmetic operations, and the rest of the instructions floating point instructions. Each load instruction takes 4 clock cycles to execute. Each arithmetic operation takes 8 clock cycle to execute. And Each floating point instructions takes 10 clock cycle to execute. The Processor is 2.3 GHz. The program contains total 8 billion instructions. How long does it take to complete the program?

Answer: CPI=0.3*4+0.2*8+0.5*10=7.8ClockcyclesperInstructions

1 instruction needs 7.8 clock cycles
8 billion instructions needs = 7.8 * 8 * 10^9 = 62.4 * 10^9 clock cycles

With 2.3 GHz processor,
2.3 * 10^9 Clock cycles generate in 1 second
62.4 * 10^9 clock cycles generate in (62.4 * 10^9) / (2.3 * 10^9) seconds. = 27.13 seconds

Question 6.

Answer: The value of the variables can be changed during the run time. However, the value of symbolic constants is fixed during the runtime.