1. Given the first four lines in the data section of an assembly program, fill in the table with the memory contents at each location whose relative address is provided. Note that only the lower 32 bits of each address are shown.

```
numbers1    db    100, -100
numbers2    dw    100, -100
number3     dd    100
number4     dq    -100
```

Show work here.

db; 100 = 64h ; -100 = 9Ch

dw; 100 = 00 64h ; -100 = FF 9Ch (H L)

dd; 100 = 0000 0064h (H1 H2 L1 L2)

dq; 100 = FF FF FF FF | FF FF FF 9C h (H1 H2 H3 H4 L1 L2 L3 L4)

	Relative address (Showing 32-bit)	Value (Hex)
1	0000 0000	64
2	0000 0001	9C
3	0000 0002	64
4	0000 0003	00
5	0000 0004	9C
6	0000 0005	FF
7	0000 0006	FF
8	0000 0007	FF
9	0000 0008	FF
10	0000 0009	FF
11	0000 000A	9C
12	0000 000B	FF
13	0000 000C	FF
14	0000 000D	FF
15	0000 000E	FF
16	0000 000F	FF
17	0000 0010	FF
18	0000 0011	FF

2. Complete the following table of Machine addition.

	Problem	Result (answer) in Hex	Correct as unsigned?	Correct as signed?	CF	OF	ZF	SF
1	64 + 9C	100	NO	Yes	1	0	0	0
2	0064 + 009C	0100	Yes	NO	0	0	0	0
3	00000064 + FFFFFF9C	1 0000 0000	NO	Yes	1	0	1	0
4	89ABCDEF01234567 + 9ABCDEF012345678	2468 ACDF 1357 9BDF	Yes	Yes	1	1	0	0

Show work here.

1. 64 + 9C = 100

Signed: 64 = 100_{10}; 9C = -100_{10}; 0_{10} or 0_{16}

2. 0064 + 009C = 0100

$0064 = 100_{10}$; $009C = 156_{10}$; 256_{10}

3. 00000064 + FFFFFF9C = 1 0000 0000

$00000064 = 100_{10}$; $FFFFFF9C = -100_{10}$; 0

4. 89ABCDEF01234567 + 9ABCDEF012345678 = 2468ACDF13579BDF

$89ABCDEF01234567 = -8,526,405,043,095,435,641_{10}$

$9ABCDEF012345678 = -7,296,712,173,568,108,936_{10}$

$2,623,536,857,045,507,039_{10}$

$= 2468\ ACDF\ 1357\ 98DF_{16}$

Part II.

1. Name the **three** types of 80x86 Assembly language statements.

Instructions	Directives	Macros

2. Give three examples for each of the following types of 80x86 Assembly Language statements.

Type	Examples		
Instruction	mov	add	syscall
directive	global	section	db

3. How many bytes will be reserved by the following statement? 13 bytes

```
message     db     "Hello, world!", 10
```

4. Write three different statements that are equivalent to the following statement.

```
message     db     "BYE!", 10
```

5 bytes

message db "BYE", "!", 10
message db "B", "Y", "E", "!", 10
message db 'BYE!', 10

5. Fill in the blanks of a partial view of an assembly listing file.

Line #	5	6	7	15	16	20
value	0A	21	0000 0011	0000 000A	0000 0014	BF 0100 0000

```
1                                       ; file name: listingfile.asm
2                                       ; display "Hello, world!" to stdout (screen)
3
4                                       section .data
5  00000000  486921 0A                      message1 db "Hi!", 10              ; String to be displayed
6  00000004  486F77206172652079-            message2 db "How are you!", 10     ; String to be displayed
6  0000000D  6F75 21 0A
7  00000011  427965210A                     message3 db "Bye!", 10             ; String to be displayed
8
9                                       global _start                           ; for ld command for linking
10
11                                      section .text
12                                      _start:
13 00000000  B801000000                  mov rax, 1                         ; sys_call id for sys_write
14 00000005  BF01000000                  mov rdi,1                          ; for stdout
15 0000000A  48BE-                       mov rsi, message1                  ; address of the string
15 0000000C  [0000000000000000]
16 00000014  BA04000000                  mov rdx,4                          ; length of the string
17 00000019  0F05                        syscall
18
19 0000001B  B801000000                  mov rax, 1                         ; sys_call id for sys_write
20 00000020  BF 0100 0000                mov rdi,1                          ; for stdout
21 00000025  48BE-                       mov rsi, message2                  ; address of the string
```