Exercise 2.2

Question: Make a diagram showing successive bytes of memory like those above to show the memory layout produced by the following data declaration:

```
Letters    DB         'ABC'
Digits     DB         1, 2, 3
Numbers    DW         6767h, 0ababh;      'h' indicates hex
More       DB         'e', 10, 'fg'
Hush       DB         5 DUP ('S'), 'H!'
Two3       DB         3 DUP (2, 3, ?)
Recurse    DB         2 DUP ('X', 3 DUP (0))
```

Answer:

Letters:	41	42	43					
Digits:	01	02	03					
Numbers:	67	67	ab	ab				
More:	65	10	66	67				
Hush:	53	53	53	53	53	48	21	
Two3:	02	03	?	02	03	?	02	03	?
Recurse:	58	00	00	00	58	00	00	00

Question: Give a single DB statement that has the same effect as the following group.

```
X          DB         5
           DB         'H'
           DW         'i'
```

Answer:

```
X          DB         5, 'Hi'
```

Question: Given the following data definitions.

```
W          DW         1234
```

```
A          DB          23
B          DB          -12
```

Which of the following mov instructions are illegal, and why?

Answer:
```
mov        W,          74          ;OK
mov        A,          74          ;OK
mov        A,          ah          ;illegal, byte can't go to word
```

Question: Fill in the contents of the specified registers as four hex digits, given the specified data definition.

Answer:
```
AWord      DW          5432h
AByte      DB          9ah
Another    DB          0bch
           ...
           mov     ax,     1234h      ;ax = 1234h
           mov     ax,     AWord      ;ax = 5432h
           mov     ax,     1234h
           mov     ah,     AByte      ;ax = 9a32h
           mov     al,     Another    ;ax = 9abch
           mov     ax,     1234h
           mov     al,     ah         ;ax = 1212h
           mov     ax,     1234h
           mov     ah,     'A'        ;ax = 4134h
           mov     ax,     1234h
           mov     ax,     'A'        ;ax = 0041h
           mov     ah,     1
           mov     al,     2          ;ax = 0102h
```

Question: What is displayed by the DOS display string call:
```
mov        dx,         OFFSET Msg
mov        ah,         9h
int        21h
```

for each of the following definitions of Msg? Assume that the cursor is at the left side of the screen before each display. Show the final position of the cursor with an underline.

a.
```
Msg        DB          'One if by land,', 13, 10
           DB          'Two if by sea.', 13, 10, '$'
```

Answer:
```
One if by land,
Two if by sea.
—
```

b.   Msg   DB   'One', 13, 10,
                                   'Two', 13, 10,
                                   'Three', 13, 10, '$'

Answer: One
              Two
              Three
              _

c.   Msg   DB   'S', 10 DUP ('h'), '!', 13, 10, '$'

Answer: Shhhhhhhhhh!

              _

Question: What is displayed by the DOS display string call:

```
mov        dx,        OFFSET Msg1
mov        ah,        9h
int        21h
mov        dx,        OFFSET Msg2
mov        ah,        9h
int        21h
```

for each of the following definitions of Msg1 and Msg2? Assume that the cursor is at the left side of the screen before each display. Show the final position of the cursor with an underline.

a.   Msg1   DB   'One if by land, $', 13, 10, '$'
              Msg2   DB   'Two if by sea.', 13, 10, '$'

Answer: One if by land,Two if by sea

              _

b.   Msg1   DB   'One if by land, ', 13, 10, '$'
              Msg2   DB   'Two if by sea.', 13, 10, '$'

Answer: One if by land,
              Two if by sea.

              _

Question: Write data definitions for Msg so that the DOS display string call:

```
mov        dx,        OFFSET Msg
mov        ah,        9h
```

```
            int              21h
```

will produce the following output:
```
      Eat
      Neat
```

Answer: Msg   DB   'Eat', 13, 10 'Neat', 13, 10, '$'