Homework 1, due February 14th 5pm CST

Handin at 1102 DCL. Slide under door if no TA present.

Important: Please type or *neatly* write your solutions. Anything we can't read will receive no credit. You must show work to receive full credit.

1. (25 points) Suppose that DS = 1310H, ES = 1102H, AX = 40H, BX = 500H, and CX = 6000H. Determine the address accessed by each of the following instructions assuming real-mode operation.

   (a) `MOV [B00H], EDX` (5 points)

   (b) `MOV [BX], EDX` (5 points)

   (c) `MOV CX[BX], EDX` (5 points)

   (d) `MOV CX[BX+AX], EDX` (5 points)

   (e) `MOV ES:[BX+AX], EDX` (5 points)

   Solution:

   (a) DS + B00H = 13100H + B00H = 13C00H

   (b) DS + BX = 13100H + 500H = 13600H

   (c) DS + CX + BX = 13100H + 6000H + 500H = 19600H

   (d) DS + CX + BX + AX = 13100H + 6000H + 500H + 40H = 19640H

   (e) ES + BX + AX = 11020H + 500H + 40H = 11560H

2. (25 points) The function `kaboom` is defined below. What will be printed to the screen when "`kaboom(100, 200, 300)`" is called?

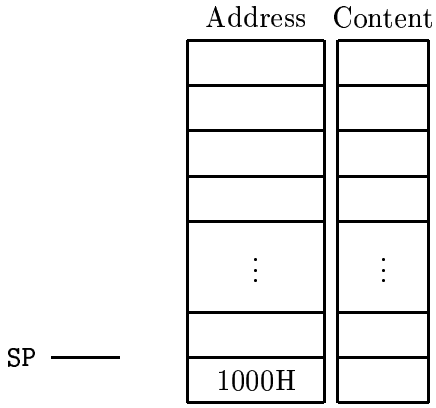
```
1  void kaboom(unsigned int x, unsigned int y, unsigned int z)
2  {
3      __asm__ ("movl %2, %0"
4               "addl %1, %0"
5               "subl %2, %1"
6               : "=a" (x)
7               : "b" (y), "c" (z)
8              );
9
10     cout << "x = " << x << endl;
11     cout << "y = " << y << endl;
12     cout << "z = " << z << endl;
13 }
```

   Solution:
   i = 500, j = 200, k = 300.

3. (25 points) Suppose that `SP = 200H`, `SS = 100H`, and `AX = AA77H`. Now suppose the instruction `PUSH AX` is run.

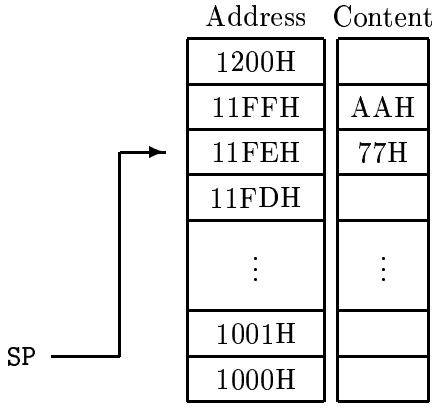
   (a) Annotate the diagram below by filling out *all* the addresses on the stack and the contents of locations in which `AX`'s value is stored. Leave the memory contents blank if they are unknown. (20 points)

   (b) What are the values of `SS` and `SP` after the instruction is finished? Also draw where `SP` points in the stack after the instruction. (5 points)



Solution:

(a) All the relevant blanks are filled out below.



(b) `SP = SP - 2 = 1FEH`. It points to `11FEH` as shown in the diagram above. `SS` remains at `100H`.

4. (25 points) Suppose we have an A/D – D/A card which uses 16-bit data and status registers. The base address of the card is set at `200H`. We also have the following registers.

Base + `AH`	Low word of A/D channel: bits $8 \ldots 15$ are bits $0 \ldots 7$ of the 18-bit data register.
Base + `BH`	High word of A/D channel: bits $0 \ldots 9$ are bits $8 \ldots 17$ of the 18-bit data register.
Base + `DH`	Write 0 into this register to start A/D conversion.
Base + `FH`	16-bit status and control register. The effects of each bit is as follows.

- Bit 15 = 1 when A/D conversion is finished (read only).
- Bit 12 = 1 to set input range to -2.5 to $+2.5$v.
- Bit 11 = 1 to set input range to -5 to $+5$v.
- Bit 10 = 1 to set input range to -10 to $+10$v.
- Bit 7 = 0 to enable interrupt, = 1 to disable.
- Bit 6 = 0 to enable DMA, = 1 to disable.
- The other bits are irrelevant.

For programmed read and write, interrupt and DMA must be disabled. You have available to you functions `hw_in()` and `hw_out()` that will input and output 16-bit unsigned *short* ints (a regular `int` is 32 bits). They function exactly like the ones seen in lab except now they deal with 16 bits instead of 8.

```
unsigned short int hw_in(unsigned int address);
void hw_out(unsigned int address, unsigned short int value);
```

Suppose that the signal voltage range is from -8v to $+8$v and the noise is ± 1v.

(a) What should be the input voltage range in the configuration? (2 points)

(b) Using your answer in part (a), write a C or C++ function to read an input voltage and print out the voltage as a float. (23 points)

Solution:

(a) -10 to +10v.

(b) The answer should be something similar to the code below.

```
1  void print_voltage()
2  {
3      unsigned int Base = 0x0200;
4      hw_out(Base + 0x000F, 0x04C0); /* initialize card */
5      hw_out(Base + 0x000D, 0x0000); /* start A/D conversion */
6      while (hw_in(Base + 0x000F) & 0x8000 == 0) {} /* wait */
7      unsigned short int loInt = hw_in(Base + 0x000A); /* read voltage */
8      unsigned short int hiInt = hw_in(Base + 0x000B);
9      unsigned int data = (hiInt << 8) | (loInt >> 8);
10     float fdata = data / 262143.0 * 20.0 - 10.0; /* convert range */
11     printf("%.4f\n", fdata);
12 }
```