Examples of IBCM Programming

Writing even very small programs for IBCM is tedious and error prone. Finding bugs in an IBCM program is even more tedious, however, so a careful systematic approach and careful hand checking at each step can save you a great deal of time and frustration in the long run.

Always start with a statement of the task and write careful pseudo code to describe the solution. Then write a carefully commented "symbolic" version of the IBCM implementation of the pseudocode. Check this version very carefully by code walkthroughs (getting a friend to help is a winner). Only then translate to the hexadecimal form, type it in (leaving the symbolic form to comment the hex form) and try to execute it. Be careful here — it's easy to forget and count

" . . . 8 9 10 11 ..."

Remember "A" comes after 9 in hexadecimal.

Below are two examples.

Each of these examples is in the "final form" after the process described above is complete. We'll walk through the process in class.

Example 1

The problem: compute the sum of the integers 1 -through- N, where N is to be read from the keyboard and the resulting sum is to be printed to the screen. Halt after printing the sum.

Pseudocode:

```
read N;
i = 1; s = 0;
while (i <=N) { s+= i; i +=1;}
print s;
```

The IBCM Code:

```
mem   locn  label op    addr        comments

C00A  00          jmp   start       skip around the variables
0000  01    i     dw    0           int i
0000  02    s     dw    0           int s
0000  03    N     dw    0           int N
0001  04    one   dw    1
0000  05    zero  dw    0
0000  06                            leave space for changes
0000  07
0000  08
0000  09
1000  0A    start readH             read N
4003  0B          store N
3004  0C          load  one         i = 1
4001  0D          store i
3005  0E          load  zero        s = 0
4002  0F          store s
3003  10    loop  load  N           if (i > N) goto xit
6001  11          sub   i
E01A  12          jmpl  xit
3002  13          load   s          s += i
5001  14          add   i
4002  15          store s
3001  16          load  i           i += 1
5004  17          add   one
4001  18          store i
C010  19          jmp   loop        goto loop
3002  1A    xit   load  s           print s
1800  1B          printH
0000  1C          halt              halt
```

Example 2

The Problem: Compute the sum of the elements of an array and print this sum on the screen (then halt). The address of the first element of the array and the size of the array are to be read from the keyboard.

Pseudocode:

```
read A; read N;
s = 0; i = 0; while (i < N) { s += a[i]; i += 1;}
print s;
```

The IBCM Code:

```
mem   locn   label  op     addr   comments

C00A  00            jmp    start  skip around the variables
0000  01     i      dw     0      int i
0000  02     s      dw     0      int s
0000  03     a      dw     0      int a[]
0000  04     n      dw     0
0000  05     zero   dw     0
0001  06     one    dw     1
5000  07     adit   dw     5000
0000  08                          leave space for changes
0000  09
1000  0A     start  readH         read array address
4003  0B            store  a
1000  0C            readH         read array size
4004  0D            store  n
3005  0E            load   zero   i = 0; s = 0;
4001  0F            store  i
4002  10            store  s
3004  11     loop   load   n      if (i >= N) goto xit
6001  12            sub    i
E020  13            jmpl   xit
D020  14            jmpe   xit
3007  15            load   adit   form the instruction to add a[i]
5003  16            add    a
5001  17            add    i
401A  18            store  doit   plant the instruction into the program
3002  19            load   s      s += a[i]
0000  1A     doit   dw     0
4002  1B            store  s
3001  1C            load   i      i += 1
5006  1D            add    one
4001  1E            store  i
C011  1F            jmp    loop   goto loop
3002  20     xit    load   s      print s
1800  21            printH
0000  22            halt
```