Homework #3

Due Date: Wednesday, March 9, 2011

Name: ....................................
1. (12 points)

The instructions below are the first 7 instructions of a Pep/8 program. Show (in hexadecimal) the contents of each byte of the user stack as it appears after the last of the instructions has been executed. Use “??” for any location where the value is unknown. You do not need to figure out the memory addresses of the stack locations.

```
lda     0x34AB,i
ldx     0x78CD,i
stbytex -4,s
subsp  -6,i
sta     4,s
adda    2,i
stbytea 0,s
```

2. (14 points)

The following program reads 32 numbers and, for each one, outputs Y it is zero and N if it is non-zero. Identify the redundant instructions in the program, that is those that could be removed without affecting the correctness of the program.

```
lda N,d
top:  deci M,d
      ldx M,d
      cpx 0,i
      breq one
      brne two
one:  charo 'Y',i
      br  endif
two:  charo 'N',i
      br  endif
endif: lda N,d
       suba 1,i
       sta N,d
       lda N,d
       brne top
stop
N:  .word 32
M:  .block 2
.end
```
3. (24 points)

Translate the following pseudocode algorithm into a complete PeP/8 assembly code program. Assume all variables mentioned are 16-bit integers.

```assembly
input (A, B, C);
while (A>0)
{
    while (B>0)
    {
        if (C>=0) output('*'); else output('+');
        B--;
    }
    output ('\n');
    A--;
}
```
4. (25 points)

Write a Pep/8 assembly code program that reads N followed by N integers. After the data is read the program outputs the (rounded) average of the largest and smallest number in the data set.

For example

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 5 -3 2 9</td>
<td>3</td>
</tr>
<tr>
<td>2 6 7</td>
<td>7</td>
</tr>
<tr>
<td>6 21 17 2 10 7 41</td>
<td>22</td>
</tr>
</tbody>
</table>
5. (25 points)

Translate the following Pep/8 program into C. Your C program need not be completely syntactically correct.

```c
ldx N,d
top:
cpx 16,i
brle exit
deci X,d
lda X,d
anda 7,i
brne skip
lda C1,d
adda 1,i
sta C1,d
lda X,d
anda 31,i
brne skip
lda C2,d
adda 1,i
sta C2,d
skip:
subx 1,i
br top
exit:
deco C1,d
deco C2,d
stop
X:    .block 2
C2:   .word 0
C1:   .word 0
N:    .word 32
.end
```