HW 16

Pointers and Functions

Name: _________________________________ Time spent: _______min

1) Given the following memory map – evaluate each item 30 pts

```
int foo;
int boo;
int zoo;
int* a_ptr;
int* b_ptr;
```

variable name	value	address
foo	0x1234	0x1000
boo	0x8000	0x2000
zoo	0x2324	0x3000
a_ptr	0x2000	0x7000
b_ptr	0x3456	0x8000

HEX

&boo

*b_ptr

a_ptr

&b_ptr

boo + b_ptr + *a_ptr + &boo (not actually allowed)

2) Fill in the memory map at the end of the following code 30 pts

*** note: this code will not compile – for illustrative purposes only ***

```
int foo;            int boo;
float zoo;          float soo;
int* a_ptr;         int*  b_ptr;
float*c_ptr;        float* d_ptr;
```

```
a_ptr = &boo;
c_ptr = a_ptr + 0x2000;
*c_ptr = 3.5 * foo;
*a_ptr = 7;
zoo = soo + boo;
d_ptr = a_ptr + c_ptr;
*d_ptr = 0x1000;
*b_ptr = boo + *a_ptr;
```

variable name	value	address
foo	5	0x1000
boo		0x2000
zoo		0x3000
soo		0x4000
a_ptr		0x5000
b_ptr		0x6000
c_ptr		0x7000
d_ptr		0x8000

3) Provide the final values after executing the following code snippit 40 pts

```
#include <stdio.h>

int fun1(int* a, int* b);
float fun2(float a, float* b);

int main(void){

   int a = 3;
   int b = 2;
   float c = 3.5;
   float d = 4.5;
   float e = 12;

   a = fun1(&a,&b);
   e = fun2(c,&d);

return 0;
}

int fun1(int* foo, int* boo ){
   int zoo;
   if (*boo > 0){
      zoo = 2**foo;
   }
   else{
      zoo = 3**foo;
   }
   *boo = zoo + *foo;
   *foo = 12;
   return zoo;
}
```

```
float fun2(float zoo, float* soo){
   zoo = 34;
   *soo *= 3*zoo;
   return *soo;
}
```

a	
b	
c	
d	
e	