CS1050 – C Programming Quiz 8 SPRING 2016

Fill out the **answer sheet** with your choices and turn the quiz in.   
  
Use the following code to answer the next several questions. IGNORE COMPILE ERRORS and assume that the address for variable “int a” is 6020, the address for pointer aPtr is 7010, the address for “int b” is 8030.  
1 #include <stdio.h>  
2 int main( void )  
3 {  
4 int a, b;  
5 int \*aPtr; ( dereferencing)  
6 a=8;  
7 printf(“%p”, &a); %p is used to print address and u also use an ampersand  
8 aPtr = &a;   
9 printf(“%d”, \*aPtr); %d decimal/ value  
10 printf(“%p”, aPtr);   
11 b = \*aPtr;  
12 printf(“%d”, b);  
13 printf(“%p”, &b);  
14 printf(“%d”, 2 \* \*aPtr);  
15 printf(“%p”, &aPtr); asking for the adress of apointer  
16 return 0;  
17 }

&a This is mostly returning the address of the pointer

1. What will the printf on line “7” display? 6020
2. What will the printf on line “9” display? 8
3. What will the printf on line “10” display? 6020
4. What will the printf on line “12” display? 8
5. What will the printf on line “13” display? 8030
6. What will the printf on line “14” display? 16
7. What will the printf on line “15” display? 7010
8. At which line of code is a pointer defined and created?

5

1. Write the function prototype for a function called **SumNum** that takes two pointers to float numbers as parameters and returns an integer. Write just the prototype – nothing else.

Int **SumNum**( float \*aptr, float \*bptr);

1. Write a function prototype for function “f” based on the line of code below assuming that variable “y” is a “char” type.  
   **f ( &y ); this is returning the location of y**  
   void f(char \*y);  
     
     
     
     
     
     
     
     
     
     
   CS1050 – C Programming Quiz 8 NAME \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   SPRING 2016 LAB SECTION \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   6. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   7. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   8. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   9. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
     
   10. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_