1.) Consider the signed primitive C types short and char on our lab’s x86_64 systems. The type short is used to represent 16-bit two’s complement numbers, and the type char is used to represent 8-bit two’s complement numbers. What output is printed after executing the following C code snippet?

```c
short s = 0xba1;
char c = (char) s;
printf("%d", c);
```

*Hint:* The printf format string character %d is used to print the decimal value of an integer type, including the primitive type char.

*Only the lower byte would be retained after the casting. The 8 bits of 0xa1 would be interpreted as a two’s complement number.*

0xa1 = 1010 0001

*Its 2’s complement value would be: -128 +32 + 1 = -95.*

2.) What is the decimal representation of the binary number 1110.1011?

14.5625

3.) For this question, please convert the decimal number 67.4375 into its binary representation. (Note: not its IEEE floating point representation; just its binary representation.)

67 = 100 0011

0.4375 = 2^{-2} + 2^{-8} + 2^{-16} = 0.0111

67.4375 = 1000011.0111

4.) Consider the following C code:

```c
// Returns the result of x+y. If negative overflow occurs, prints an error message
// and exits.
int add(int x, int y)
{
    int result = x + y;

    if( condition ){
        printf("Negative overflow occurred\n");
        exit(1);
    }
    else{
        return result;
    }
}
```

Provide C code that would replace condition in the code in order to detect when negative overflow occurs.
\[ x < 0 \&\& y < 0 \&\& \text{result} \geq 0 \]

\[ \text{result} \geq 0 \text{ because } T_{min} + T_{min} = 0. \]

5.) Consider the following C code snippet:

```c
int arr[4];
int *ptr = &arr[1];
arr[0] = 7;
*(ptr+1) = 100;
*(arr+2) = 42;
int val = arr[2];
*ptr = 3;
```

Assume that \texttt{ints} are stored in 4 bytes and pointers are stored in 8 bytes. The memory addresses (in decimal) associated with the first byte of each of the variables is as follows:

- arr: 100
- ptr: 116
- val: 124

Indicate what values would be stored at the end of the code:

- arr[0] = 7
- arr[1] = 3
- arr[2] = 42
- arr[3] = ??
- ptr = 104
- *ptr = 3
- val = 42