1. Convert -6.625 to 32-bit IEEE single precision format

2. Convert 123.457 to 32 32-bit IEEE single precision format.

3. What floating-point number is represented by **0x41BA0000**.

4. Assume we are multiplying the unsigned integers **1011 X 1011**. Trace the values of the multiplicand, multiplier, and result at every step. (We are not covering this algorithm until Monday April 11).

5. The swap function below exchanges the two double values pointed to by **x** and **y**. Write **swap** as an ARM assembly language function. Full credit for the <u>most concise</u> version.

```
void swap(double *x, double *y) {
     double tmp = *x;
     *x = *y;
     *y = tmp;
}
```

6. Write a recursive C function that implements the declaration below. **popcount** counts the number one bits in the binary representation of its argument. For example, **popcount(30)** is 4 because 30 in binary is 11110, which has four one bits.

```
extern int popcount(unsigned int n);
```

7. Write **popcount** as an ARM assembly language function.

8. Consider the logic function with three inputs **A**, **B**, **C** and one output **Out**. **Out** should be 1 when exactly two inputs are 1.

   a. Draw the truth table for this function.
   b. Write the sum-of-products logic equation for this function.
   c. Minimize the logic equations
   d. Draw the circuit diagram for the logic equation.

9. Write a C function scale that takes a factor and multiplies each item in the array by the factor.

```
extern void scale(double factor, double [] vec, int n);
```

10. Make sure you understand the four areas of program memory; code, global data, stack, and heap and how memory is allocated for each.

11. Static function local variables in C are allocated in/on _____ memory.

12. Local variables in C are allocated in/on _____ memory.

13. Memory allocated using `malloc` is _____ memory.

14. What does the `-g` flag on the gcc compiler do?

15. What does the `-S` flag on the gcc compiler do?

16. What does the `-o` flag on the gcc compiler do?

17. What does the `-O3` flag on the gcc compiler do?

18. What does the `-c` flag on the gcc compiler do?

19. What program do we use to reverse engineer machine code files?

20. How many bytes is a C **double**?

21. Briefly describe what a *memory leak* is?

22. Consider the following C program. Why might it have a segmentation fault?

```
#include <stdio.h>
int *seven() {
    int x = 7;
    return &x;
}

int main() {
    int *y = seven();
    printf("%d\n", *y);
}
```

23. The following variation of the program seems to work OK. Why?

```
#include <stdio.h>

int *seven() {
    static int x = 7;
    return &x;
}

int main() {
    int *y = seven();
    printf("%d\n", *y);
}
```

24. Write a function **rev** that takes an unsigned integer **x** and reverses the bits in **x**. Use bit operations only, don't use strings or arrays.
    a. Modify the **add** function in **adder.c** we wrote to call **rev**.

25. There is a simple fix to the **add** function in **adder.c** file that does not need to reverse the bits. What is it?