1. (1 points) Consider the following simple program. Give the values printed by the printf function.

```
#include <stdio.h>
int main() {
    int i,j,k;
    i=2;
    k = j = i -= 3;
    printf("%d %d %d
", i, j, k);
}
```

output: -1 -1 -1

```
#include <stdio.h>
int main() {
    int i, j, k; i = 7; j = 7; k = -10;
    printf("%d", k > i > j);
    i = 1; j = 2; k = 3;
    printf("%d", i < j == j < k);
}
```

output: 0 1

2. (1 points) Write the following as a C-expression:

\[
\frac{x^2 + y^2}{\left(\frac{x}{y}\right)^3}
\]

\[(x*x + y*y) / ((x*x*x) / (y*y*y)) \quad OR \quad (\text{pow}(x,2) + \text{pow}(y,2)) / \text{pow}((x/y), 3)\]

\text{pow}() \text{ is a function of math.h library}

3. (2 points) Consider the following expression which is evaluated by the C-compiler based on the operator precedence and the operator associativity of C-language (precedence and associativity are given in the lecture note and in the text book). With the help of the table of precedence and associativity (on page 63 of the text book), give an EQUVALENT fully parenthesized expression which will not require any operator precedence or associativity to evaluate the expression. Give the values of the printf function.

```
int a, b, c, d, e, f;
    a = b = c = d = e = f = 5;
    a = b += --a + c++ - f-- - d + e * f / c;
```
printf("%d %d %d %d\n", a, b, c, f);

**Priority Statement (Precedence Statement):**

\[ a = b + (-a) + (c++) - (f--) - d + ((e * f) / c); \]

**Final values:**

\[ a = 7 \]
\[ b = 7 \]
\[ c = 6 \]
\[ f = 4 \]

4. (2 points) Besides the binary representation we have learned from the class, another very useful way to represent a number is the hexadecimal representation. Here is an introduction from Wikipedia:

Hexadecimal is a positional numeral system with a radix, or base, of 16. It uses sixteen distinct symbols, most often the symbols 0–9 to represent values zero to nine, and A, B, C, D, E, F (or alternatively a–f) to represent values ten to fifteen. Hexadecimal numerals are widely used by computer systems designers and programmers. Several different notations are used to represent hexadecimal constants in computing languages; the prefix "0x" is widespread due to its use in Unix and C (and related operating systems and languages). Alternatively, some authors denote hexadecimal values using a suffix or subscript. For example, one could write 0x2AF3 or 2AF3_{16}, depending on the choice of notation.

As an example, the hexadecimal number 2AF3_{16} can be converted to an equivalent decimal representation. Observe that 2AF3_{16} is equal to a sum of \((2000_{16} + A00_{16} + F0_{16} + 3_{16})\), by decomposing the numeral into a series of place value terms. Converting each term to decimal, one can further write:

\[ 2AF3_{16} = (2_{16} \times 16^3) + (A_{16} \times 16^2) + (F_{16} \times 16^1) + (3_{16} \times 16^0) \]
\[ = (2 \times 4096) + (10 \times 256) + (15 \times 16) + (3 \times 1) \]
\[ = 10995 \]

Each hexadecimal digit represents four binary digits (bits), and the primary use of hexadecimal notation is a human-friendly representation of binary-coded values in computing and digital electronics. One hexadecimal digit represents a nibble, which is half of an octet or byte (8 bits). For example, byte values can range from 0 to 255 (decimal), but may be more conveniently represented as two hexadecimal digits in the range 00 to FF. Hexadecimal is also commonly used to represent computer memory addresses.

To convert a decimal number to the hexadecimal representation, we can do the following:

1. First convert the decimal number to the binary representation;
2. In the binary representation, from right to left, separate every four digits into a group;
3. Convert each group of four digits to a decimal number;
(4) Replace the decimal numbers with hexadecimal symbols and write out the final hexadecimal representation.

For example, convert $387922_{10}$ to hexadecimal representation:

$$387922_{10} = 01011110101101010010_2$$
$$= 5 \quad 14 \quad 11 \quad 5 \quad 2_{10}$$
$$= 5 \quad E \quad B \quad 5 \quad 2_{16}$$
$$= 5EB52_{16}$$

Convert decimal number $111_{10}$ to hexadecimal representation. Show the details of each step.

$$111_{10} = (0110 \quad 111)_2$$
$$= (6 \quad F)_{16}$$

5. (1 point) Consider the following code with scanf() function. Assume that the user who runs this code wants to assign 10, 4 and 5 to height, width and depth respectively. What are you going to see if you run this code and want to enter the mentioned values? (Write everything that is printed and entered)

```c
#include <stdio.h>
int main() {
    int height, width, depth;

    printf("Enter the height:");
    scanf("%d", &height);

    printf("Enter the width:");
    scanf("%d", &width);

    printf("Enter the depth:");
    scanf("%d", &depth);

    printf("The volume is: %d", height*width*depth);
}
```

Output:

```
Enter the height: 10
Enter the width: 4
Enter the depth: 5
The volume is: 200
```