1. S = 1 / ((1-f) + f/s)
    a. S = 2.5 s = 4
       2.5 = 1 / ((1-f) + f/4)
       1 = 2.5 - 2.5f + 0.625f
       -1.5 = -1.875f
       f = 0.8

    b. s = 6, f = 0.8
       S = 1 / ((1-0.8) + 0.8 / 6)
       S = 1 / (0.2 + 0.133)
       S = 1 / 0.333
       S = 3x speedup

2. Hexadecimal and Binary from int

```c
void hex_bin(int x){

    printf("The number in hexadecimal: 0x");

    int i, bits = sizeof(x) * 8;
    for (i = bits / 4 - 1; i >= 0; i--) {
        int nibble = (x >> (i * 4)) & 0xF;
        if (nibble < 10) {
            printf("%d", nibble);
        } else {
            printf("%c", 'A' + nibble - 10);
        }
    }

    printf("\nThe number in binary: ");

    for (i = bits - 1; i >= 0; i--) {
            printf("%d", (x & (1 << i)) ? 1 : 0);
        if (i % 4 == 0)
            printf(" ");
    }

    printf("\n");

}
```

3. Short, long, double

```c
#include <stdio.h>

typedef unsigned char *byte_pointer;

void show_bytes(byte_pointer start, size_t len) {
    int i;
    for (i = 0; i < len; i++)
        printf(" %.2x", start[i]);
    printf("\n");
}

void show_int(int x) {
    show_bytes((byte_pointer) &x, sizeof(int));
}

void show_float(float x) {
    show_bytes((byte_pointer) &x, sizeof(float));
}

void show_pointer(void *x) {
    show_bytes((byte_pointer) &x, sizeof(void *));
}

void show_short(short x) {
    show_bytes((byte_pointer) &x, sizeof(short));
}

void show_long(long x) {
    show_bytes((byte_pointer) &x, sizeof(long));
}

void show_double(double x){
    show_bytes((byte_pointer) &x, sizeof(double));
}
```