# In Class work for Lecture 3/29/2017

Objective:  Learn about program optimization: Observe the speed up by manual inlining, code_motion, reduction in strength

1. Observer the effect of inlining by doing it manually
    1.1. Compile inline_base.c with no optimization
    1.2. Manually inline 7 instances of “calculate_index”
        1.2.1. Compile this version with no optimization
2. Now, further inline 1 instance of “set_row”
    2.1. Compile this version with no optimization
3. Now, further use “code_motion” to:
    3.1. move any instances of i*COLUMNS outside of the inner loops
4. Use reduction in strength to transform the multiplications to additions
5. Finally fully optimize your manually optimized code
    5.1. Compare it to fully optimizing the original

| Condition | How long did it take to run the program? |
|---|---|
| gcc -o inline_base inline_base.c | |
| (calculate_index) gcc -o inline_manual_01 inline_manual_01.c | |
| (set_row) gcc -o inline_manual_02 inline_manual_02.c | |
| (code_motion) gcc -o inline_manual_03 inline_manual_03.c | |
| (reduction in strength) gcc -o inline_manual_04 inline_manual_04.c | |
| gcc -O3 -o inline_manual_04_max inline_manual_04.c | |
| gcc -O3 -o inline_base_max inline_base.c | |

6. Were you able to improve performance by manually optimizing your code?
7. Were you able to beat the automatic optimizations?