Homework assignments are to be completed individually. Hand-written submissions are fine, but they must be readable. Due at the beginning of class. Total points: 100, 5% of course grade

1. [15 Points] Induction Variables. Perform induction variable detection, strength reduction and induction variable elimination on Figure 9.3 in book.

2. [15 Points] SSA. Translate Figure 9.5 into SSA. Perform copy propagation on SSA.


   \[
   \begin{align*}
   a &= \text{read()} \\
   b &= \text{read()} \\
   z &= \text{read()} \\
   w &= \text{read()} \\
   x &= a + b \\
   y &= a + b \\
   j &= 0 \\
   \text{loop:} \\
   &\quad z = a \ast b \\
   &\quad \text{if} (z > j) \text{ goto L1} \\
   &\quad w = a \ast b \\
   &\quad \text{goto L2} \\
   \text{L1:} \\
   &\quad w = a \ast b \\
   \text{L2:} \\
   &\quad x = a + b \\
   &\quad y = x \\
   &\quad j = j + 1 \\
   &\quad \text{if} (j < 10) \text{ goto loop} \\
   \text{print} \ w,x,y,z
   \end{align*}
   \]

   (a) Perform pessimistic global value numbering.
   (b) Perform optimistic global value numbering.
   (c) Transform the SSA code based on the optimistic global value numbering results. How can you rewrite phi functions such as "x_1 = phi(x_0, x_0)"? After doing so, what optimization on SSA (hint problem 2 in this HW) should be performed? Perform it.
   (d) Rewrite the resulting code in 3-address code.
4. [20 points] Data dependence analysis and unimodular transformations

```c
for (i=0; i<N; i++) {
    for (j=0; j<M; j++) {
        B[j+1][i] = x + y * i;
    }
}
```

(a) For the above program, what is direction vector for the output dependence? (Hint: Recall that (\*, <), (\*, =), and (\*, >) are not legal dependence vectors.)

(b) For the above program, what is the distance vector for the flow dependence?

(c) What is the unimodular transformation matrix that specifies a permutation of the i and j loops in the program for problem 4?

(d) Is the problem 4 loop fully permutable? Why or why not?

(e) Which loop carries each of the dependences? What is a possible parallelization strategy for the above loop?

5. [15 points] Loop Fission and the Kelly and Pugh Transformation Framework

(a) Show whether loop fission is legal or illegal for the following program using the K&P transformation framework.

```c
for (i=0; i<N; i++) {
    A[i] = ... ;
    ... = A[i - 1];
}
```

(b) Show whether loop fission is legal or illegal for the following program using the K&P transformation framework.

```c
for (i=0; i<N; i++) {
    A[i] = ... ;
    ... = A[i + 2];
}
```

6. [15 points] Loop transformations and Fourier Motzkin. Skew the loop to make it permutable and then permute the loop. Write the transformed code.

```c
for (i=0; i<N; i++) {
    for (j=0; j<i; j++) {
```
A[i][j] = A[i-1][j] + A[i-1][j+1];