

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, 3.3% of course grade

1. [15 Points] Induction Variables. Perform induction variable detection, strength reduction and induction variable elimination on Figure 9.5 in book.
2. [15 Points] SSA. Translate Figure 9.3 into SSA. Perform copy propagation and dead code elimination on SSA.
3. [20 Points] Value Numbering.

```

    a = read()
    b = read()
    z = read()
    w = read()
    x = a - b
    y = a + b
    j = 0
loop:
    z = a * b
    w = z
    if (w==0) goto L2
    w = a * b
L2:
    x = a + b
    j = j + 1
    if (z < j) goto loop
print w,x,y,z

```

- (a) Perform pessimistic global value numbering.
 - (b) Transform the SSA code based on the optimistic global value numbering results. After doing so, what optimization on SSA (hint problem 2 in this HW) should be performed? Perform it.
 - (c) Rewrite the resulting code in 3-address code.
4. [20 points] Data dependence analysis and unimodular transformations
 - (a) For the below loop, what is the direction vector for the output dependences between writes to $A[i]$? (Hint: Recall that $(*, <)$, $(*, =)$, and $(*, >)$ are not legal dependence vectors.)
 - (b) For the below loop, 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 loop below fully permutable? Why or why not?
- (e) Which loop carries each of the dependences? What is a possible parallelization strategy for the above loop?

```

for (i=0; i<N; i++) {
  for (j=3; j<M; j++) {
    A[i+1] = B[ j-3][ i+2 ] - 99;
    B[ j-1][ i ] = sin(i * j);
  }
}

```

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.

```

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.

```

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.

```

for (i=1; i<N; i++) {
  for (j=1; j<(i+1); j++) {
    A[ i ][ j ] = A[i-2,j] + A[i-1,j+1];
  }
}

```