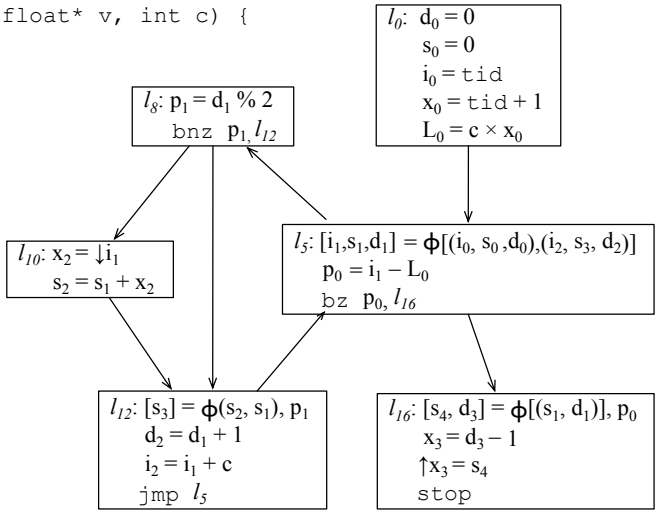
DCC888 – Divergence Analysis

Name: _____ ID: _____

1. This question refers to the program below, which computes the sum of each superior column of a matrix. The matrix is passed to this kernel as a linear vector. The GSA representation of this program is given on the right:

```
1 __global__ void sumTriangle(float* m, float* v, int c) {
2   if (T_id < c) {
3     int d = 0;
4     float sum = 0.0F;
5     int L = (T_id + 1) * c;
6     for (int i = T_id; i < L; i += c) {
7       if (d % 2) {
8         sum += m[i];
9       }
10      d += 1;
11    }
12    v[d - 1] = sum;
13  }
14}
```



(a) Draw the graph of dependencies of `sumTriangle`. Use solid edges to represent data dependencies, and dashed edges to represent control dependencies.

(b) Given the graph that you just drew above, which variables in the GSA representation of `sumTriangle` are divergent?

(c) Does your divergence analysis mark the loop in line 6 as divergent or not? How do you extract this information from the results of the divergence analysis?

(d) Is the result of your divergence analysis accurate for the loop at line 6? Is this loop truly divergent in practice? If the result of the analysis is accurate, then it is called a *true positive*, otherwise it is called a *false positive*.

(e) What about the branch in line 7. Does your divergence analysis marks it as divergent or not? How do you infer this information from the outcomes produced by the analysis? Is this result accurate?

2. Consider the program below, which averages the columns of a matrix. Answer all the items from the previous question with regard to this new program. Naturally, you do not have to answer item 1(e), given that the new kernel `avgSquare` does not have a branch inside the loop.

```
1 __global__ void avgSquare(float* m, float* v, int c) {
2  if (T_id < c) {
3    int d = 0;
4    float sum = 0.0F;
5    int L = T_id + c * c;
6    for (int i = T_id; i < L; i += c) {
7        sum += m[i];
8        d += 1;
9    }
10   v[tid] = sum / d;
11 }
12}
```