**Problem definition:**

Task Communication Graph (TCG) is a set of tasks (*V*) and a set of communication activities (*E*) where *V* = {*v0, v1, … vn*} and *E* = {*e0, e1, … em*}. The edge is directed whose source and destinations are denoted as *S*(*ek*) = *vi*, *D*(*ek*) = *vj*, which means the source of *ek* is *vi* and the destination is *vj*. The weight of an edge, *W*(*ek*) denotes the amount communication on the edge.

The other input is a set of processors (*P*) and its topology is *X* by *Y* mesh. An element *px,y* in *P* denotes a processor at position (*x*, *y*) in the mesh where 0 ≤ *x* < *X* and 0 ≤ *y* < *Y*. A processor can only communicate with its adjacent processors and the communication overhead between adjacent processors is fixed. Therefore, the communication overhead between two arbitrary processors, *pa,b* and *pc,b* is proportional to their Manhattan distance, i.e. *H*(*pa,b*, *pc,b*) = |*a-c*| + |*b-d*|.

The problem is to find a mapping of *V* and *P* such that the communication overhead is minimized. The mapping is formally defined as *M*(*vi*) = *px,y*. The communication overhead is for all *ek* in *E*, the summation of *W*(*ek*) • *H*(*M*(*S*(*ek*), *M*(*D*(*ek*))).

**Input file format:**

The format of an input file is <source task (*S*(*ek*))> <destination task (*D*(*ek*))> <weight *W*(*ek*)> for all edges in *E*.

For example, for the following TCG,

20

10

10

the input file is

0 1 20

0 2 10

1 2 10

**Output file format:**

The output should be mapping of tasks and processors. A processor is identified by its coordination. The format is <task (*vi*)> <x coordination of the mapped processor> <y coordination of the mapped processor>

For example, for the mapping of the following mapping,

0, 0

0, 1

1, 0

1, 1

the output file is

0 0 0

1 0 1

2 1 1