

Graph is a very common data structure in computer algorithm. A graph $G = (V,E)$ is composed of a set of vertices, or nodes, and a set edges between the nodes. For a complete and weighted graph $G = (V,E,W)$, each pair of nodes is connected by an edge with a weight. The weights W can be represented by an n -by- n matrix, where n is the number of nodes $|V|$ in the graph G .

In 2-dimensional plane, a node can be represented by its x coordinate and y coordinate, and the weights of a complete and weighted graph can be calculated as distances between all pairs of nodes. Your goal is to provide a program to input some graph through some set of nodes, and output their distance matrix.

Your program should provide two classes `Node` and `Graph`. The `Node` class must contain:

1. Two private data members of type integer representing the **coordinates** x and y .
2. A **default constructor** initializing the coordinates (x,y) to $(0,0)$.
3. Corresponding set and get functions for the data members.
4. **Overloaded** stream **insertion** ($<<$) and **extraction** ($>>$) operators.

The `Graph` class must contain:

1. A private data member of type `size_t` representing the **number of nodes** in the graph.
2. A private data member of type `Node*` representing the **set of nodes** V .
3. A private data member of type `unsigned int**` representing the **weights** W .
4. A **default constructor** initializing all the data members to zero or null pointer.
5. An **overloaded constructor** initializing the graph object through an array of nodes (`Node*`) and the size of the array (`size_t`).
6. A **destructor** reclaiming all the claimed resources.
7. Corresponding set and get functions for the data members.
8. **Overloaded** stream **insertion** ($<<$) and **extraction** ($>>$) operators.

Requirement: Use the sample main function to complete your program. Provide two classes `Node` and `Graph` satisfying all of the above conditions and separate the interface (`Node.h`, `Graph.h`) and implementation (`Node.cpp`, `Graph.cpp`).

Prohibited: C-style code.

Input

The integer in the first line indicates the number of cases. Each case contains an integer n (the number of nodes), followed by $2n$ integers (the coordinates of the n cities).

Output

For each case, output the weight matrix. The Euclidean distance is rounded down to an integer. The field width of each distance is 2, and adjacent columns are separated by one space. Adjacent rows are separated by a newline stream manipulator.

Sample Input

```
1
5
-1 2
3 1
4 3
5 -4
-5 0
```

Sample Output

```
0  4  5  8  4
4  0  2  5  8
5  2  0  7  9
8  5  7  0 10
4  8  9 10  0
```