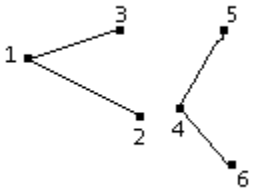
Cheaper Processing Units

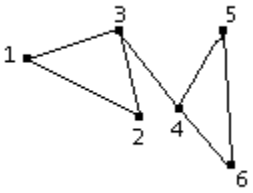
Filename: cpu

United Computer Fabrication (UCF) is designing a new microprocessor, the Knightron XL, that will eclipse even the latest offerings from the other two major CPU makers. UCF plans on using this new offering to take over the lucrative gamer and enthusiast markets. To do this, UCF plans to make the Knightron XL as affordable as possible and undersell the other two manufacturers (the less a gamer has to spend on his CPU, the more he or she can spend on the rest of the rig).

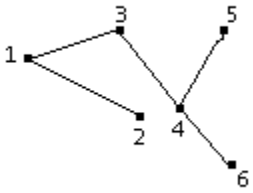
There are many potential designs for the Knightron XL. Each design incorporates several subsystems on a silicon die that must be connected to each other with gold wire traces. Even such a tiny amount of gold wire can cost a great deal as hundreds of thousands of processors are produced. So UCF's first step in reducing cost is to minimize the amount of gold wire used to connect the subsystems. There are many different ways to properly connect the subsystems. See the diagrams below:



System 1

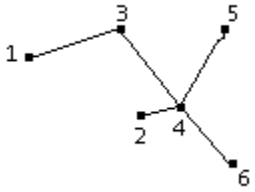


System 2



System 3

System 1 is no good because there is no way for subsystems 1, 2, or 3 to communicate with subsystems 4, 5, or 6. This design is not properly connected. System 2 is properly connected, but there are redundant connections between some of the subsystems, making the design inefficient. If we remove the connections between subsystems 2 and 3 and subsystems 5 and 6, as in System 3, the amount of wire used has been greatly reduced. However, this still is not the most efficient solution. Note that subsystem 2 is much closer to subsystem 4 than it is to subsystem 1. If we remove the connection between subsystems 1 and 2 and add a connection between subsystems 2 and 4, as in the figure below, we obtain an even more efficient design.



The Problem:

UCF wants to be able to rapidly identify Knightron XL designs that require the smallest amount of wire to connect the subsystems. Your job as head of the software development at UCF is to write a program that will automatically analyze CPU designs and output the smallest length of gold wire needed to properly connect the subsystems.

The Input:

There will be multiple designs to consider. Input will begin with a single integer, *n,* on the first line, indicating how many designs are being considered. Each design will begin with a single integer, *s* ($1 \le s \le 20$), on a line by itself, indicating how many subsystems are in this particular design. This will be followed by *s* lines, each line representing one of the subsystems (subsystem 1 is on the first line, subsystem 2 is on the second, etc.). On each of the *s* lines, there will be *s* non-negative integers, each representing the amount of wire (in micrometers) required to connect from this subsystem to another subsystem (the first integer represents subsystem 1, the second represents subsystem 2, etc.). The length of wire from any subsystem to itself is zero and the length of wire between two connected subsystems is both positive and symmetric (the same amount of wire would be required to go either direction between the two subsystems).

The Output:

For each design, print "`Design x:` " where *x* represents the number of the design (starting at 1), followed by the minimum amount of gold wire, *w*, needed to properly connect the subsystems together in the format "*w* `micrometers`" as shown in the Sample Output.

Sample Input:

```
2
3
0 1 2
1 0 4
2 4 0
5
0 3 2 4 1
3 0 1 2 5
2 1 0 7 1
4 2 7 0 3
1 5 1 3 0
```

Sample Output:

```
Design 1: 3 micrometers
Design 2: 5 micrometers
```