Problem B: Modular NN

|  |  |
| --- | --- |
| Author:  **Nikola Nedeljkovic** | *Implementation and analysis:*  ***Janko Sustersic, Nikola Nedeljkovic*** |

Statement:

Due to the recent popularity of *Deep learning* newcountries are starting to look like *Neural Networks.* That is, the countries are being built *deep* with many layers, each layer possibly having many *cities*. They also have *one* entry, and *one* exit point. There are exactly layers each having cities. Let us look at two adjacent layers and . Each city from layer is connected to each city from layer with the travelling cost , and each pair of adjacent layers has the same cost in between their cities as any other pair (they just stacked the same layers, as usual). Also, the travelling costs to each city from layer are same for all cities in that is is the same for and fixed . *Doctor G.* needs to speed up his *GPU* computations for this country so he asks you to find the number of paths he can take from *entry* to *exit* point such that his travelling cost is divisible by given number .

***Input:***

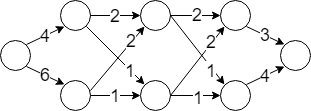
The first line of input contains – the number of cities in each layer, L – the number of layers, and. Second, third and fourth line contain integers denoting costs from *entry* point to first layer, costs between adjacent layers as described above, and costs from last layer to *exit* point.

***Output:***

Output a single integer, the number of paths *Doctor G.* can take which have total cost divisible by modulo

|  |  |
| --- | --- |
| ***Example input:*** | ***Example output:*** |
| 2 3 13  4 6  2 1  3 4 | 2 |

***Example explanation:***

******

This is a country with layers, each layer having 2 cities. Paths , and are the only paths having total cost divisible by 13. Notice that input edges for layer cities have same cost, and that they are same for all layers.

***Constraints:***

***Time and memory limit: 1s / 64 MB***

Solution and analysis:

# TODO