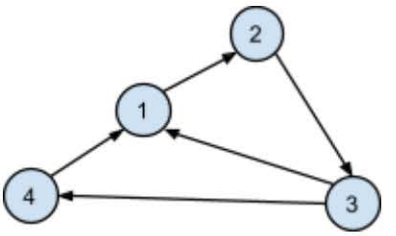
# American Computer Science League

## 2020-2021 ● Contest 4: Graphs ● Senior Division

**PROBLEM:** Given a directed graph, find all simple paths of a specified length that start at each of the vertices in the graph. Output the sum of all the paths, where each path is considered as a number. The graphs will contain at most 9 vertices, with labels 1 through 9.

For example, in the graph below the simple paths of length 2 are as follows: From vertex 1, the path is 123. From vertex 2, paths are 231 and 234. From vertex 3, the paths are 341 and 312. From vertex 4, the only simple path is 412. The sum is 123 + 231 + 234 + 341 + 312 + 412 = 1653.

In the graph below, the simple paths of length 3 are 1234, 2341, 3412, and 4123.




**INPUT:** Your program will receive a single line of input, each will contain a number, *n*, the length of the paths to be found, followed by a list of 2-character strings giving all of the directed edges in the graph.

**OUTPUT:** Print the sum of all paths of length *n*.

**SAMPLE INPUT:**

```
2 12 23 34 41 31
3 12 23 34 41 13 32
4 67 75 54 12 13 23 31 34 41 56 45
3 34 45 56 63 64 61 13
2 12 21 13 15 53 33
```

**SAMPLE OUTPUT:**

```
1653
15242
356313
37651
581
```

**TEST INPUT:**

```
2 12 31 41 42 43 45 51 63 64 56 16
3 12 13 22 23 24 34 42 98 71 87 17 96 67
4 12 14 21 24 25 32 41 43 59 65 91 87 76 95
3 11 12 14 15 23 25 31 43 45 51 52 68 79 87 89
2 55 77 45 54
```

**TEST OUTPUT:**

```
8478
74349
754366
59578
0
```

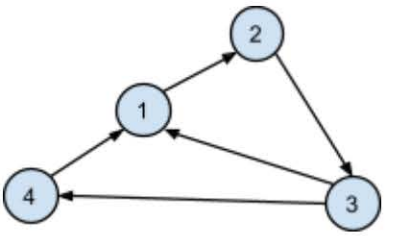
# American Computer Science League

2020-2021 ● Contest 4: Graphs ● Senior Division

**问题:** 给定一个有向图， 找出从图中每个顶点出发的所有指定长度的简单路径。输出所有路径之和，将其中每条路径都看作一个数。图中最多包含 9 个 顶 点 ， 标 记 为 1 - 9 。

例如，在下图中，长度为 2 的简单路径如下：从顶点 1 开始，路径为 123。从顶点 2 开始，路径为 231 和 234。 从顶点 3 开始， 路径为 341 和 312。从顶点 4 开始，唯一的简单路径为 412。所以路径之和为 123 + 231 + 234 + 341 + 312 + 412 = 1653。

在下图中，长度为 3 的简单路径为 1234、 2341、 3412 和 4123。




**输入:** 你将会接收到一行数据，每行都包含一个数字 ***n***，表示需要寻找的路径的长度，紧接着是一系列包含 2 个字符的字符串，表示图中所有的有向边。

**输出:** 打印输出所有长度为 ***n*** 的路径之和。

**样本输入:**

```
2 12 23 34 41 31
3 12 23 34 41 13 32
4 67 75 54 12 13 23 31 34 41 56 45
3 34 45 56 63 64 61 13
2 12 21 13 15 53 33
```

**预期输出:**

```
1653
15242
356313
37651
581
```

**测试输入:**

```
2 12 31 41 42 43 45 51 63 64 56 16
3 12 13 22 23 24 34 42 98 71 87 17 96 67
4 12 14 21 24 25 32 41 43 59 65 91 87 76 95
3 11 12 14 15 23 25 31 43 45 51 52 68 79 87 89
2 55 77 45 54
```

**预期输出:**

```
8478
74349
754366
59578
0
```