# American Computer Science League

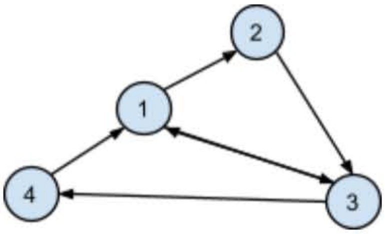
## 2020-2021 • Contest 4: Graphs • Junior Division

**PROBLEM:** Given a directed graph, create its adjacency matrix in order to output one of the following:

1. Find and print the sum of the number of cycles of length 1 and cycles of length 2.
2. Find the maximum number of edges starting at any specific vertex. If there is a tie, choose the vertex that is first numerically. Print the sum of all edges that start at that vertex.
3. Find and print the total number of paths of length 2 in the entire graph.

For example, in the graph below, the answer to each characteristic stated above is:

1. 1 - There are 0 cycles of length 1 since there is no edge that starts and ends with the same vertex. There is one cycle of length 2 which exists because 13 and 31 are both edges. The sum is 1.
2. 25 - Vertices 1 and 3 each have a maximum of 2 edges starting there. Therefore, use starting vertex 1 since it is first numerically. The sum of all of the edges that start at vertex 1 is 12 + 13 = 25.
3. 10 - By inspection, the paths of length 2 are 123, 131, 134, 231, 234, 313, 312, 341, 412, and 413. The total is 10.



**INPUT:** Your program will receive a single line of input, each will contain a number from 1-3 to indicate which of the above 3 characteristics to print followed by a list of 2-character strings giving all of the directed edges inthe graph. For example, the string "31" says there is a directed edge from vertex 3 to vertex 1. Graphs will have no more than 9 vertices.

**OUTPUT:** Print the result of the specified characteristic (1-3) for the corresponding graph that was input.

**SAMPLE INPUT:**

```
2 12 13 23 31 34 41
1 12 23 34 11 21 32 45 53 95 43 99 29 91
3 12 23 34 41 31 52 45 61 14 21 33 55 13 54 32 56 36
1 12 11 33 34 43 55 52 41 31 25 88 79 98 45 13 42 87 35 51 21 14 78
2 12 11 33 34 43 55 52 41 31 25 88 79 98 45 13 42 87 35 51 21 14 78
```

**SAMPLE OUTPUT:**

```
 25
 5
 49
 10
 50
```

**TEST INPUT:**

```
1 12 31 41 42 43 45 51 63 64 56 16
2 12 13 22 23 24 34 42 98 71 87 17 96 67
3 12 14 21 24 25 32 41 43 59 65 91 87 76 95
2 11 12 14 15 23 25 31 43 45 51 52 68 79 87 89
3 55 77 45 54
```

**TEST OUTPUT:**

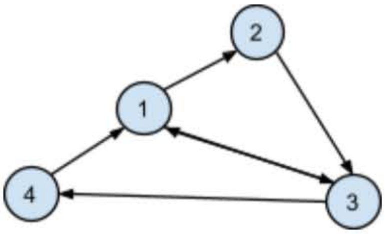
```
0
42
24
52
6
```

**问题:** 给定一个有向图，创建它的邻接矩阵，使得最终输出为以下特征之一：

1. 找到并打印输出长度为 1 和 2 的环的数量总和。
2. 找到从任意一个特定顶点出发的边的最大数量。如果边数相等，选择数值靠前的顶点。打印输出从该顶点出发的所有边的和。
3. 找到并打印输出整个图中长度为 2 的路径数量总和。

例如，在下图中，满足上述每个特征的结果为：

1. 1 - 因为没有一条边的出发顶点和结束顶点相同，所以没有长度为 1 的环。因为 13 和 31 都是边，所以存在一个长度为 2 的环。因此总和为 1。
2. 25 - 从顶点 1 和顶点 3 作为出发顶点最多有 2 条边。因为从数值上看数字 1 靠前，所以将顶点 1 作为出发顶点。因此从顶点 1 开始的所有边的总和为 12 + 13 =25.
3. 10 - 经过观察，长度为 2 的路径有 123, 131, 134, 231, 234, 313, 312, 341, 412 和 413，总数为 10。



**输入：** 你将会接收到一行数据，每行都包含一个 1-3 的数字，表示要输出上述 3 个特征中的哪一个，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。比如，字符串 “31” 表示一条从顶点 3 到顶点 1 的有向边。图不超过 9 个顶点。

**输出:** 打印输出对应图中选定特征 (1-3) 的结果。

**样本输入:**

```
2 12 13 23 31 34 41
1 12 23 34 11 21 32 45 53 95 43 99 29 91
3 12 23 34 41 31 52 45 61 14 21 33 55 13 54 32 56 36
1 12 11 33 34 43 55 52 41 31 25 88 79 98 45 13 42 87 35 51 21 14 78
2 12 11 33 34 43 55 52 41 31 25 88 79 98 45 13 42 87 35 51 21 14 78
```

**预期输出:**

```
 25
 5
 49
 10
 50
```

测试输入:

```
1 12 31 41 42 43 45 51 63 64 56 16
2 12 13 22 23 24 34 42 98 71 87 17 96 67
3 12 14 21 24 25 32 41 43 59 65 91 87 76 95
2 11 12 14 15 23 25 31 43 45 51 52 68 79 87 89
3 55 77 45 54
```

预期输出:

```
 0
 42
 24
 52
 6
```