Memory Limit: 1024 MB
Time Limit: 5 s

Graph Parkour (100 points)

Introduction

You are a parkour master, someone who climbs buildings and jumps off ledges all while living to tell about it. Unfortunately you were in a bad accident a few weeks ago where you broke both of your arms. The doctors say that you aren’t allowed to parkour like you used to, but you don’t want to hear what they say, so you set out about planning your parkour path through the city. Unfortunately, because of your broken arms, you won’t be able to climb up while on your path, and you will only be able descend. Your friend, another parkour master, has been cataloging the area and has been keeping track of all the buildings, how tall they are, and the best ways to get between them. Luckily enough for you, he’s stored all this in a graph. In your friend’s graph each vertex is a building and each edge is a way to move between the buildings.

As you want to get as much use out of each run as possible, you want to find the longest path between buildings where you are only descending from building to building. As you are a parkour master, you can descend as far as you want and not worry about injury.

Input Specifications

You will be given a string that describes the graph, vertices in the graph will be separated by semicolons “;” and vertexes will be described as “Name:Height-Edge1,Edge2,...,EdgeN” where “Name” is the name of the vertex, “Height” is the height of the building that is represented by the vertex, and “EdgeN” is the name of the vertex that this vertex has an edge to.

Output Specifications

You should output the length of the longest path of descending buildings. In the example, the output would be 3 because you can make the path “B->A->C”

Sample Input/Output

Input

```
A:12-B,C;B:14-A;C:5-B,A;
```

Output

```
3
```

Explanation

The longest path is B->A->C

Input

```
A:4-D,E;C:0-B,A,D;B:5-D,A,C,E;E:2-D,A,B,C;D:4-;
```

Output

4

Explanation

The longest path is B->A->E->C