



Magic Spells



by Dalimil

Problem

Submissions

Leaderboard

Discussions

You are battling a powerful dark wizard. He casts his spells from a distance, giving you only a few seconds to react and conjure your counterspells. For a counterspell to be effective, you must first identify what kind of spell you are dealing with.

The wizard uses scrolls to conjure his spells, and sometimes he uses some of his generic spells that restore his stamina. In that case, you will be able to extract the name of the scroll from the spell. Then you need to find out how similar this new spell is to the spell formulas written in your spell journal.

Spend some time reviewing the locked code in your editor, and complete the body of the *counterspell* function.

Check [Dynamic cast](#) to get an idea of how to solve this challenge.

Input Format

The wizard will read t scrolls, which are hidden from you.

Every time he casts a spell, it's passed as an argument to your *counterspell* function.

Constraints

- $1 \leq t \leq 100$
- $1 \leq |s| \leq 1000$, where s is a scroll name.
- Each scroll name, s , consists of uppercase and lowercase letters.

Output Format

After identifying the given spell, print its name and power.

If it is a generic spell, find a subsequence of letters that are contained in both the spell name and your spell journal. Among all such subsequences, find and print the length of the longest one on a new line.

Sample Input

```
3
fire 5
AquaVitae 999 AruTaVae
frost 7
```

Sample Output

```
Fireball: 5
6
Frostbite: 7
```

Explanation

Fireball and *Frostbite* are common spell types.

AquaVitae is not, and when you compare it with *AruTaVae* in your spell journal, you get a sequence: *AuaVae*

Submissions: 2942

Max Score: 40

Difficulty: Hard

Rate This Challenge:

[More](#)

Need Help? Get advice from the [discussion forum](#) for this challenge. Or check out the [environments page](#)

Current Buffer (saved locally, editable)  

C++



```
1 ▶ #include <iostream>
4 using namespace std;
5
6 class Spell {
7     private:
8         string scrollName;
9     public:
10        Spell(): scrollName("") { }
11        Spell(string name): scrollName(name) { }
12        virtual ~Spell() { }
13        string revealScrollName() {
14            return scrollName;
15        }
16 };
17
18 class Fireball : public Spell {
19     private: int power;
20     public:
21        Fireball(int power): power(power) { }
22        void revealFirepower(){
23            cout << "Fireball: " << power << endl;
24        }
25 };
26
27 class Frostbite : public Spell {
28     private: int power;
29     public:
30        Frostbite(int power): power(power) { }
31        void revealFrostpower(){
32            cout << "Frostbite: " << power << endl;
33        }
34 };
35
36 class Thunderstorm : public Spell {
37     private: int power;
38     public:
39        Thunderstorm(int power): power(power) { }
40        void revealThunderpower(){
41            cout << "Thunderstorm: " << power << endl;
42        }
43 };
44
45 class Waterbolt : public Spell {
46     private: int power;
47     public:
48        Waterbolt(int power): power(power) { }
49        void revealWaterpower(){
50            cout << "Waterbolt: " << power << endl;
51        }
52 };
53
54 class SpellJournal {
55     public:
56        static string journal;
57        static string read() {
58            return journal;
59        }
60 }
```

```

60 };
61 string SpellJournal::journal = "";
62
63 void counterspell(Spell *spell) {
64     /* Enter your code here */
65
66     Fireball *foo1 = dynamic_cast<Fireball*>(spell);
67     Frostbite *foo2 = dynamic_cast<Frostbite*>(spell);
68     Waterbolt *foo3 = dynamic_cast<Waterbolt*>(spell);
69     Thunderstorm *foo4 = dynamic_cast<Thunderstorm*>(spell);
70     if(foo1 != NULL) {
71         foo1->revealFirepower();
72     }
73     else if(foo2 != NULL) {
74         foo2->revealFrostpower();
75     }
76     else if(foo3 != NULL) {
77         foo3->revealWaterpower();
78     }
79     else if(foo4 != NULL) {
80         foo4->revealThunderpower();
81     }
82     else {
83         string X = spell->revealScrollName();
84         string Y = SpellJournal::read();
85         int m = X.size();
86         int n = Y.size();
87         int L[m+1][n+1];
88         int i, j;
89         for (i = 0; i <= m; i++) {
90             for (j = 0; j <= n; j++) {
91                 if (i == 0 || j == 0)
92                     L[i][j] = 0;
93                 else if (X[i-1] == Y[j-1])
94                     L[i][j] = L[i-1][j-1] + 1;
95                 else
96                     L[i][j] = max(L[i-1][j], L[i][j-1]);
97             }
98         }
99         cout << L[m][n] << endl;
100     }
101 }
102
103
104 class Wizard {
105     public:
106     Spell *cast() {
107         Spell *spell;
108         string s; cin >> s;
109         int power; cin >> power;
110         if(s == "fire") {
111             spell = new Fireball(power);
112         }
113         else if(s == "frost") {
114             spell = new Frostbite(power);
115         }
116         else if(s == "water") {
117             spell = new Waterbolt(power);
118         }
119         else if(s == "thunder") {
120             spell = new Thunderstorm(power);
121         }
122         else {
123             spell = new Spell(s);
124             cin >> SpellJournal::journal;
125         }
126         return spell;
127     }
128 };
129
130 int main() {
131     int T;

```

```
132     cin >> T;
133     Wizard Arawn;
134     while(T--) {
135         Spell *spell = Arawn.cast();
136         counterspell(spell);
137     }
138     return 0;
139 }
```

Line: 101 Col: 2

[Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

Congrats, you solved this challenge!

✓ Test Case #0

✓ Test Case #3

✓ Test Case #6

✓ Test Case #9

✓ Test Case #1

✓ Test Case #4

✓ Test Case #7

✓ Test Case #2

✓ Test Case #5

✓ Test Case #8

Next Challenge

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)