



UVA10409

Life is not easy. Sometimes it is beyond your control. Now, as contestants of ACM ICPC, you might be just tasting the bitter of life. But don't worry! Do not look only on the dark side of life, but look also on the bright side. Life may be an enjoyable game of chance, like throwing dice. Do or die! Then, at last, you might be able to find the route to victory.

"Life is not easy."，人生往往超出我們的掌握現在，作為ACM ICPC的參賽者，您可能只是在品嚐生活的痛苦。但是不用擔心！不要只看人生的黑暗面，也要看光明面。人生可能是一種令人愉快的機會遊戲，例如擲骰子"Do or die!"。說不定可以找到通往勝利的途徑！

This problem comes from a game using a die. By the way, do you know a die? It has nothing to do with "death." A die is a cubic object with six faces, each of which represents a different number from one to six and is marked with the corresponding number of spots. Since it is usually used in pair, "a die" is a rarely used word. You might have heard a famous phrase "the die is cast," though.

此問題來自骰子遊戲。"Do you know a die?"此處的"die"與死亡無關，而是指一般的立方體骰子，每個面代表一到六個不同的數字。順帶一提，"a die"是一個很少使用的詞。不過，您可能會聽過一個名言："the die is cast"

題目

4

When a game starts, a die stands still on a flat table. During the game, the die is tumbled in all directions by the dealer. You will win the game if you can predict the number seen on the top face at the time when the die stops tumbling.

遊戲開始時，骰子會在平台上靜止不動。
在遊戲中，主持人將骰子向各個方向滾動。
如果您可以預測骰子停止滾動時在頂面上看到的數字，則您將贏得比賽。

Now you are requested to write a program that simulates the rolling of a die. For simplicity, we assume that the die neither slips nor jumps but just rolls on the table in four directions, that is, north, east, south, and west. At the beginning of every game, the dealer puts the die at the center of the table and adjusts its direction so that the numbers one, two, and three are seen on the top, north, and west faces, respectively. For the other three faces, we do not explicitly specify anything but tell you the golden rule: the sum of the numbers on any pair of opposite faces is always seven.

現在，要求您編寫一個模擬骰子滾動的程式。為簡單起見，我們假設骰子既不滑動也不跳躍，而只是在四個方向(東，南，西，北)上滾動。

在每局遊戲開始時，主持人將骰子放在桌子的中央並調整其方向，以便分別在頂面、北面、西面上看到數字1、2、3。

對於其他三個面，我們沒有明確指定任何內容，但會告訴您一條黃金法則：任何一對相對的面的數字總和始終為7。

題目

6

Your program should accept a sequence of commands, each of which is either “north”, “east”, “south”, or “west”. A “north” command tumbles the die down to north, that is, the top face becomes the new north, the north becomes the new bottom, and so on. More precisely, the die is rotated around its north bottom edge to the north direction and the rotation angle is 90 degrees. Other commands also tumble the die accordingly to their own directions. Your program should calculate the number finally shown on the top after performing the commands in the sequence. Note that the table is sufficiently large and the die never falls off during the game.

您的程式應接受一系列指令，指令為東"east"，南"south"，西"west"，北"north"。例如"north"指令將骰子向下滾動到北，即頂面變為新的北，北變為新的底，依此類推。其他指令也會根據自己的方向滾動骰子。執行順序中的指令後，您的程式應計算最終顯示在頂部的數字。請注意，桌子足夠大，骰子在遊戲中不會掉落或損壞。

輸入與輸出

7

Input : The input consists of one or more command sequences, each of which corresponds to a single game. The first line of a command sequence contains a positive integer, representing the number of the following command lines in the sequence. You may assume that this number is less than or equal to 1024. A line containing a zero indicates the end of the input. Each command line includes a command that is one of 'north', 'east', 'south', and 'west'. You may assume that no white space occurs in any lines.

Output : For each command sequence, output one line containing solely the number on the top face at the time when the game is finished.

輸入：輸入第一行包含一個正整數 n ($n \leq 1024$)，代表指令序列的數量。如果 $n = 0$ 代表輸入結束。每個指令只包含東"east"、"south"、"west"、"north"。

輸出：對於每個指令序列，在遊戲結束時輸出頂面上的數字。

Input

1
north
3
north
east
south
0

Output

5
1

Step 1：輸入測資

變數	備註
N	骰子滾動次數
n、e、w、s、t、b	北東西南上下所代表數字
str	骰子滾動方向

```
5  
6  
7  
8  
9  
10  
int N;  
while (cin>>N,N!=0) {  
    int e=4,s=5,w=3,n=2,t=1,b=6;  
    for(int i=0;i<N;i++){  
        string str;  
        cin>>str;
```

Step 2：依照骰子滾動方向 更改數值

變數	備註
N	骰子滾動次數
n、e、w、s、t、b	北東西南上下所代表數字
str	骰子滾動方向

```
12 if(str[0]=='n'){
13     temp=t;
14     t=s;
15     s=b;
16     b=n;
17     n=temp;
18 }
19 if(str[0]=='s'){
20     temp=t;
21     t=n;
22     n=b;
23     b=s;
24     s=temp;
25 }
26 if(str[0]=='e'){
27     temp=t;
28     t=w;
29     w=b;
30     b=e;
31     e=temp;
32 }
33 if(str[0]=='w'){
34     temp=t;
35     t=e;
36     e=b;
37     b=w;
38     w=temp;
39 }
```

Step 3：輸出上方的數值

變數	備註
N	骰子滾動次數
n、e、w、s、t、b	北東西南上下所代表數字
str	骰子滾動方向

```
41 | cout<<t<<endl;
```

完整程式碼

12

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main() {
5      int N;
6      while (cin >> N, N != 0) {
7          int e=4, s=5, w=3, n=2, t=1, b=6;
8          for (int i=0; i<N; i++) {
9              string str;
10             cin >> str;
11             int temp;
12             if (str[0] == 'n') {
13                 temp=t;
14                 t=s;
15                 s=b;
16                 b=n;
17                 n=temp;
18             }
19             if (str[0] == 's') {
20                 temp=t;
21                 t=n;
22                 n=b;
```

```
23                 b=s;
24                 s=temp;
25             }
26             if (str[0] == 'e') {
27                 temp=t;
28                 t=w;
29                 w=b;
30                 b=e;
31                 e=temp;
32             }
33             if (str[0] == 'w') {
34                 temp=t;
35                 t=e;
36                 e=b;
37                 b=w;
38                 w=temp;
39             }
40         }
41         cout << t << endl;
42     }
43     return 0;
44 }
```

資料來源

13

英文題目 : <https://vjudge.net/problem/UVA-10409>

中文題目 : <https://zerojudge.tw/ShowProblem?problemid=e516>

THANK YOU