Graph algorithms form a very important part of computer science and have a lineage that goes back at least to Euler and the famous *Seven Bridges of Königsberg* problem. Many optimization problems involve determining efficient methods for reasoning about graphs.

This problem involves determining a route for a postal worker so that all mail is delivered while the postal worker walks a minimal distance, so as to rest weary legs.

Given a sequence of streets (connecting given intersections) you are to write a program that determines the minimal cost tour that traverses every street at least once. The tour must begin and end at the same intersection.

The "real-life" analogy concerns a postal worker who parks a truck at an intersection and then walks all streets on the postal delivery route (delivering mail) and returns to the truck to continue with the next route.

The cost of traversing a street is a function of the length of the street (there is a cost associated with delivering mail to houses and with walking even if no delivery occurs).

In this problem the number of streets that meet at a given intersection is called the *degree* of the intersection. There will be at most two intersections with odd degree. All other intersections will have even degree, i.e., an even number of streets meeting at that intersection.

Input

The input consists of a sequence of one or more postal routes. A route is composed of a sequence of street names (strings), one per line, and is terminated by the string '`deadend`' which is NOT part of the route. The first and last letters of each street name specify the two intersections for that street, the length of the street name indicates the cost of traversing the street. All street names will consist of lowercase alphabetic characters.

For example, the name `foo` indicates a street with intersections `f` and `o` of length 3, and the name `computer` indicates a street with intersections `c` and `r` of length 8. No street name will have the same first and last letter and there will be at most one street directly connecting any two intersections. As specified, the number of intersections with odd degree in a postal route will be at most two. In each postal route there will be a path between all intersections, i.e., the intersections are connected.

Output

For each postal route the output should consist of the cost of the minimal tour that visits all streets at least once. The minimal tour costs should be output in the order corresponding to the input postal routes.

Sample Input

```
one
two
three
deadend
mit
dartmouth
linkoping
tasmania
york
emory
cornell
duke
kaunas
hildesheim
concord
arkansas
williams
glasgow
deadend
```

Sample Output

```
11
114
```