Problem 3: Who Are You?

The Charitable Donations Registry (CDR) maintains a central list of all people who have made charitable donations in North America. Unfortunately, as people may use different versions of their names when contributing to different charities, this list may contain several entries for the same person. As the multiple donation requests ensuing from such duplicate entries can annoy donors to the point where they will not give any money to any charity anymore, the CDR is making an effort to remove such duplicates.

The first step in this process is to determine which name in the list is most similar to a given name (and hence may indicate the same person). This will be done using a normalized name similarity-score $S(N1, N2) = match(N1, N2)/length(N1)$, where $match(N1, N2)$ denotes the largest possible score associated with a chain of matched name-elements $N1$ and $N2$ and $length(N1)$ is the number of name-elements in $N1$. The score of two matched name-elements is 1 if the elements are identical, 0.5 if either is an initial that matches the first letter of the other, and 0 otherwise. The score of a chain of matched elements is the sum of the individual matches. For example, the highest-scoring chain for "Todd Wareham" and "Harold T. Wareham" is ⟨ "Todd" – "T.", "Wareham" – "Wareham" ⟩, which yields similarity-value $(0.5 + 1)/2 = 1.5/2 = 0.75$., whereas the highest-scoring chain for "Herbert W. Hoover" and "C. W. Ford Maddox H." is ⟨ "W." – "W.", "Hoover" – "H." ⟩, which yields similarity-value $(1 + 0.5)/3 = 0.5$. Note that elements must preserve temporal order in match-chains and cannot be switched around; for example, the highest-scoring chain for "H. Todd Wareham" and "H. T. Wareham" is ⟨ "H." – "H.", "Todd" – "T.", "Wareham" – "Wareham" ⟩ with similarity-value $(1 + 0.5 + 1)/3 = 0.83$ but the highest-scoring chain for "H. Todd Wareham" and "T. H. Wareham" is ⟨ "H." – "H.", "Wareham" – "Wareham" ⟩ with similarity-value $(1 + 1)/3 = 0.67$.

Write a program which, given a name N and a list of names L, computes and outputs $S(N, N')$ for each N' in L, as well as the name N' in L with the highest value $S(N, N')$. Your input will be an $(1 + |L|)$-line textfile, in which the first line is the given name and the remaining $|L|$ lines are the names in L (one name per line). You may assume that all input files are formatted correctly and that there is exactly one N' with maximum value for $S(N, N')$ in any given L.

Sample input #1 (available as file "test3a.dat"):

```
Todd Wareham
Harold T. Wareham
H. Wareham
H. T. Wareham
H. Todd Wareham
Todd Wilkie
T. Wareham
Harold Wareham
```

Sample output #1:

```
>>> match("Todd Wareham","Harold T. Wareham") = 0.750000
>>> match("Todd Wareham","H. Wareham") = 0.500000
>>> match("Todd Wareham","H. T. Wareham") = 0.750000
>>> match("Todd Wareham","H. Todd Wareham") = 1.000000
>>> match("Todd Wareham","Todd Wilkie") = 0.500000
>>> match("Todd Wareham","T. Wareham") = 0.750000
>>> match("Todd Wareham","Harold Wareham") = 0.500000
Best match to "Todd Wareham" is "H. Todd Wareham"
```

Sample input #2 (available as file "test3b.dat"):

```
H. Todd Wareham
Todd Wareham
Harold T. Wareham
H. Wareham
H. T. Wareham
T. H. Wareham
T. Wareham
Harold Wareham
Todd Wilkie
```

Sample output #2:

```
>>> match("H. Todd Wareham","Todd Wareham") = 0.666667
>>> match("H. Todd Wareham","Harold T. Wareham") = 0.666667
>>> match("H. Todd Wareham","H. Wareham") = 0.666667
>>> match("H. Todd Wareham","H. T. Wareham") = 0.833333
>>> match("H. Todd Wareham","T. H. Wareham") = 0.666667
>>> match("H. Todd Wareham","T. Wareham") = 0.500000
>>> match("H. Todd Wareham","Harold Wareham") = 0.500000
>>> match("H. Todd Wareham","Todd Wilkie") = 0.333333
Best match to "H. Todd Wareham" is "H. T. Wareham"
```