UVa Online Judge

12962 Average Reuse Distance

Calculating the *reuse distance* of a certain memory location is a very common technique in computer architecture, since it offers knowledge about the locality of programs that is independent of the architecture of the machine.

The reuse distance of a reference to a memory address in a program is defined by the number of different addresses that have been referenced since the last reference to the same address. In other words, if a program references the addresses ‘a’, ‘b’, ‘c’, ‘d’, and ‘a’ (we will denote addresses by lowercase letters), the reuse distance of the last reference to ‘a’ (with respect to the previous reference to ‘a’) is 3, since references to 3 different addresses have been performed. Similarly, if the references in a program are ‘a’, ‘b’, ‘b’, and ‘a’, the reuse distance of the last reference to ‘a’ will be 1.

Your task is to write a program that calculates the average reuse distance of a sequence of references. You have to consider that the first reference to an address does not have reuse distance, and therefore is not computed.

Input

The program input consists of several lines, each one with a sequence of characters from ‘a’ to ‘z’ representing references to different memory locations. The input terminates with a line with the string ‘0’.

Output

For each line, your program must print the average reuse distance of the performed references with four decimal digits, or ‘NaN’ (Not a Number) if no reference is reused. The forth decimal digit must be rounded to the nearest value; for example, 1.000049 must be printed as 1.0000; and 1.00005 and 1.00006 must be printed as 1.0001.

Hint: in C++, use:

```
#include <iomanip>
...
cout << setprecision(4) << fixed;
```

Sample Input

```
a
ab
aa
aba
abba
abcbdac
abcdbac
problembyalbertorosbardisa
0
```

Sample Output

```
NaN
NaN
0.0000
1.0000
0.5000
2.3333
2.6667
4.5385
```