1. Kevin and Stuart want to play the 'The Minion Game'.

**Game Rules :**

Both players are given the same string, S.

Both players have to make substrings using the letters of the string S.

Stuart has to make words starting with consonants.

Kevin has to make words starting with vowels.

The game ends when both players have made all possible substrings.

**Scoring**

A player gets +1 point for each occurrence of the substring in the string S.

**For Example:**

String S = BANANA

Kevin's vowel beginning word = ANA

Here, ANA occurs twice in BANANA. Hence, Kevin will get 2 Points.

For better understanding, see the image below:



Your task is to determine the winner of the game and their score.

**Input Format**

A single line of input containing the string S .

**Note:** The string  S will contain only uppercase letters: [A-Z] .

**Constraints**

0 < len(S) <= 106

**Output Format**

Print one line: the name of the winner and their score separated by a space.

If the game is a draw, print Draw.

**Sample Input**

BANANA

**Sample Output**

Stuart 12

1. The itertools module standardizes a core set of fast, memory efficient tools that are useful by themselves or in combination. Together, they form an iterator algebra making it possible to construct specialized tools succinctly and efficiently in pure Python.

To read more about the functions in this module, check out their [documentation here](https://docs.python.org/2/library/itertools.html).

You are given a list of N lowercase English letters. For a given integer K , you can select any K  indices (assume 1-based indexing) with a uniform probability from the list.

Find the probability that at least one of the K indices selected will contain the letter: 'a'.

**Input Format**

The input consists of three lines. The first line contains the integer N, denoting the length of the list. The next line consists of N  space-separated lowercase English letters, denoting the elements of the list.

The third and the last line of input contains the integer K, denoting the number of indices to be selected.

**Output Format**

Output a single line consisting of the probability that at least one of the K indices selected contains the letter:'a'.

**Note**: The answer must be correct up to 3 decimal places.

**Constraints**

1<= N <= 10

1<=K <= N

All the letters in the list are lowercase English letters.

**Sample Input**

4

a a c d

2

**Sample Output**

0.8333

3.  Tieu owns a pizza restaurant and he manages it in his own way. While in a normal restaurant, a customer is served by following the first-come, first-served rule, Tieu simply minimizes the average waiting time of his customers. So he gets to decide who is served first, regardless of how sooner or later a person comes.

Different kinds of pizzas take different amounts of time to cook. Also, once he starts cooking a pizza, he cannot cook another pizza until the first pizza is completely cooked. Let's say we have three customers who come at time t=0, t=1, & t=2 respectively, and the time needed to cook their pizzas is 3, 9, & 6 respectively. If Tieu applies first-come, first-served rule, then the waiting time of three customers is 3, 11, & 16 respectively. The average waiting time in this case is (3 + 11 + 16) / 3 = 10. This is not an optimized solution. After serving the first customer at time t=3, Tieu can choose to serve the third customer. In that case, the waiting time will be 3, 7, & 17 respectively. Hence the average waiting time is (3 + 7 + 17) / 3 = 9.

Help Tieu achieve the minimum average waiting time. For the sake of simplicity, just find the integer part of the minimum average waiting time.

**Input Format**

* The first line contains an integer N, which is the number of customers.
* In the next N lines, the ith line contains two space separated numbers Ti and Li. Ti is the time when ith customer order a pizza, and Li is the time required to cook that pizza.
* The ith customer is not the customer arriving at the  ith  arrival time.

**Output Format**

* Display the integer part of the minimum average waiting time.

**Constraints**

* 1 ≤ N ≤ 105
* 0 ≤ Ti ≤ 109
* 1 ≤ Li ≤ 109

**Note**

* The waiting time is calculated as the difference between the time a customer orders pizza (the time at which they enter the shop) and the time she is served.
* Cook does not know about the future orders.

Sample Input #00

3

0 3

1 9

2 6

Sample Output #00

9

Sample Input #01

3

0 3

1 9

2 5

Sample Output #01

8