## Задача – Прецизност

Автор: Ивайло Кенов

### Не знам за вас, но повечето програмисти много мразят математическите константи – типично са много дълги, с повечко цифри и изключително трудни за запаметяване. Обикновените дроби, от друга страна, са в пъти по-лесни за запомняне и дават достатъчно добра прецизност при работа. Примерно не е ли по-лесно да запомните числото Пи като обикновена дроб 22/7 вместо десетичната 3,14159265358979323846264338327? Единственият потенциален проблем в случая – дробното представяне е вярно само до втората цифра след запетаята. Една идея по-добър пример е 355/133 – то има коректност до шестата цифра.

Да работим с безкрайни десетични дроби – не е кеф, затова трябва вие да измислите добър метод за приблизително пресмятане на обикновените им дроби. Вашата задача е при зададени стриктно по-малка от **1** положителна десетична дроб **Number** и максимален знаменател **MaxDenominator**, да намерите най-добрата по прецизност обикновена дроб по следните правила:

1. Имаме дроб **Fraction** = **Nominator** / **Denominator**, където **0 <= Nominator < Denominator**.
2. Нека **Decimal** е десетичното представяне на **Fraction** (крайно или безкрайно).
3. Нека **Digits** са броя цифри след десетичната запетая в **Number**. Ако в **Number** имаме незначещи нули, то те също се броят към **Digits**.
4. Ако **Decimal** е безкрайна или съдържа повече цифри от **Digits**, взимат се предвид само първите **Digits** цифри след десетичната запетая на **Decimal**. Останалите се премахват без закръгляне.
5. Ако **Digits** е повече от цифрите след десетичната запетая в **Decimal**, добавете незначещи нули след **Decimal**, докато цифрите не се изравнят с **Digits**.
6. Най-добрата по прецизност **Precision** обикновена дроб е тази, която има най-много еднакви цифри от началото до края на числото с **Number.** Броим до първа различна цифра. Например числата **3.14286** and **3.14156** имат еднаква част **3.14** и тяхната **Precision** е **три**, защото три цифри са равни.

От вас се иска да намерите най-добрата (с най-голяма прецизност **Precision**) обикновената дроб на **Number**, чийто знаменател е по-малък или равен на **MaxDenominator**. Математически казано – намерете обикновената дроб **Fraction** = **Nominator** / **Denominator**, където **1 <= Denominator <= MaxDenominator** и **0 <= Nominator < Denominator**. Ако има няколко варианта верни решения, които са с еднаква прецизност, изберете този с най-малък знаменател **Denominator**. Ако отново има няколко варианта, изберете този с най-малък числител **Nominator**.

### Вход

### Входът се чете от конзолата.

### На първия ред ще прочетете максималния знаменател MaxDenominator.

### На втория ред ще прочетете числото Number.

Входните данни винаги ще бъдат валидни и в описания формат.

### Изход

### Изходът се печата на конзолата.

### На първия ред от изхода отпечатайте обикновената дроб в стил “Nominator/Denominator”

На втория ред от изхода отпечатайте прецизността, с която **Number** и обикновената дроб си съвпадат.

### Ограничения

* **MaxDenominator** ще бъде между 1 и 100 000, включително.
* **Number** съдържа от 3 до 50 символа, включително.
* **Number** ще бъде число от типа “0.ddddd...”, където d е цифра межу 0 и 9, включително.
* Разрешеното време за изпълнение на програмата е 0.2 секунди. Лимит на паметта: 16 МБ.

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 42  0.141592658 | 1/7  3 |
| **Вход** | **Изход** |
| 3  0.1337 | 0/1  1 |
| **Вход** | **Изход** |
| 80000  0.1234567891011121314151617181920 | 10/81  8 |
| **Вход** | **Изход** |
| 1000  0.42 | 3/7  3 |
| **Вход** | **Изход** |
| 100  0.420 | 21/50  4 |
| **Вход** | **Изход** |
| 115  0.141592658 | 16/113  7 |