

Fast IO 2



Biblioteka C przygotowana przez projekt GNU zawiera zbiór funkcji umożliwiających wczytywanie danych bez blokowania strumienia. Nazwy tych funkcji zakończone są sufiksem "_unlocked". Zdarza się, że różnica w szybkości wczytywania danych pomiędzy tymi funkcjami, a ich odpowiednikami blokującymi strumień jest na tyle duża, że umożliwia rozwiązanie zadania nawet przy użyciu algorytmu o gorszej złożoności. Taka sytuacja nie powinna mieć miejsca, dlatego też koordynatorzy AlgoLigi opracowali program, który analizuje nadesłany kod i wyodrębnia z niego definicje makr. Twoim zadaniem jest sprawdzenie czy podane makro zawiera w sobie podciąg znaków **unlocked**. Podciąg ten nie musi być spójny.

Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita n ($0 < n < 10^5$) oznaczająca liczbę makr. W kolejnych n liniach znajdują się definicje makr. Każda definicja może mieć jedną z dwóch postaci:

1. **NAZWA = wartość** jak nietrudno się domyślić w tym wypadku każde wystąpienie nazwy makra jest zastępowane jego wartością. Nazwa makra składa się wyłącznie z wielkich liter alfabetu angielskiego, zaś jej długość nie przekracza 4 znaków. Wartość z kolei składa się wyłącznie z małych liter alfabetu angielskiego i jest nie dłuższa niż 100 znaków.
2. **NAZWA1 = NAZWA2 NAZWA3** postać tego typu określa, że makro **NAZWA1** powstaje z połączenia wyników makr **NAZWA2** i **NAZWA3**. W przypadku takiej definicji zawsze łączone są 2 makra.

W ostatnim wierszu znajduje się nazwa makra, które należy zbadać pod kątem występowania podciągu **unlocked**. Gwarantujemy, że wszystkie wymagane do rozwiązania definicje makr wystąpią na wejściu.

Wyjście

Jeżeli badane makro zawiera szukany podciąg to należy wypisać **Fast I/O**, w przeciwnym wypadku **I/O**.

Przykład

Wejście

```
5
KAN = TU JE
TU = aubncldo
JE = M Y
M = ck
Y = uuueuuuuuuu
KAN
```

Wyjście

