

## Zadanie I

### Domeny

Serwery DNS (Domain Name System) umożliwiają konwersję internetowych adresów symbolicznych na numeryczne adresy protokołu IP. Serwery te zorganizowane są w rozproszony geograficznie, hierarchiczny system.

Skupimy się na funkcjonalności pojedynczego serwera, zarządzającego olbrzymią liczbą domen. Domeny pamiętane są w postaci rekordów (wpisów) składających się z: nazwy symbolicznej oraz odpowiadającego jej adresu IP. Serwer obsługuje następujące operacje:

- **INSERT** nazwa adres - dodaje nowy rekord lub uaktualnia istniejący,
- **FIND** nazwa - odnajduje adres skojarzony z podaną nazwą i wypisują ją na wyjście. Jeśli szukanego wpisu nie ma w bazie, operacja wypisuje **ERROR**,
- **DELETE** nazwa - usuwa wpis odpowiadający podanej nazwie z rejestru. Jeśli nazwy nie ma w bazie, operacja wypisuje **ERROR**, jeśli jest, wypisuje na wyjście skojarzony z nią adres.

Twoim zadaniem jest zaimplementowanie powyższej funkcjonalności w oparciu o **tablicę haszującą**. Zużycie pamięci Twojego programu powinno być liniowe względem liczby istniejących wpisów. Oczekiwany, zamortyzowany czas obsługi pojedynczej operacji powinien być stały.

### Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą  $z$  ( $1 \leq z \leq 2 \cdot 10^9$ ) – liczbę zestawów danych, których opisy występują kolejno po sobie. Opis jednego zestawu jest następujący:

Opis jednego zestawu jest następujący: Pierwsza linia zawiera liczbę naturalną  $n$  ( $1 \leq n \leq 2000000$ ) oznaczającą liczbę operacji do wykonania. Kolejne  $n$  linii zawiera: kod operacji oraz odpowiednie dla operacji argumenty oddzielone spacją. Argumentami są (w zależności od operacji): nazwa (maksymalnie 20-znakowy napis składający się z małych liter alfabetu angielskiego i kropek), adres (cztery liczby całkowite z zakresu  $0 \dots 255$  oddzielone kropkami). Możesz założyć, że liczba istniejących wpisów w bazie w żadnym momencie czasowym nie przekracza pół miliona. Program powinien wykorzystywać obszar pamięci, którego wielkość jest proporcjonalna do liczby przechowywanych rekordów.

### Wyjście

Dla każdej kolejnej operacji **FIND** należy wypisać linie zawierającą poszukiwany adres (albo **ERROR**, jeżeli wpisu nie ma w bazie). Dla każdej kolejnej operacji **DELETE** należy wypisać dotychczasowy adres skojarzony z podaną nazwą (albo **ERROR**, jeżeli wpisu nie było w bazie).

**Dostępna pamięć: w zależności od testu 2-164MB**

## Przykład

Dla danych wejściowych:

```
1
10
INSERT tcs.uj.edu.pl 149.156.75.211
INSERT localhost 127.0.0.1
INSERT example.com 1.2.3.4
FIND tcs.uj.edu.pl
FIND example.com
FIND example.com
INSERT example.com 1.2.3.5
FIND example.com
DELETE localhost
DELETE localhost
```

Poprawną odpowiedzią jest:

```
149.156.75.211
1.2.3.4
ERROR
1.2.3.5
127.0.0.1
ERROR
```