

Zahlensuche

Vor vielen, vielen Jahren als die Welt noch schwarzweiß war und das Telefon eine Scheibe hatte, gab es dicke Bücher in welchen jeder Bürger der Stadt mit Namen und Telefonnummer verzeichnet waren. Noch heute findet man vereinzelt solche Relikte einer vergangenen Zeit.

So hat auch Florian bei seiner Großmutter ein altes, verstaubtes Telefonbuch gefunden. Er schaut sich die Nummern an und sieht, dass sein Vater auch in dem Telefonbuch verzeichnet ist, sein Bruder jedoch nicht. Schnell hat Florian analysiert, dass in diesem alten Buch nur Menschen verzeichnet sind, die über 40 sind.

Nun möchte er herausfinden, wer von den Leuten, die er in seinem Handy gespeichert hat, auch in das Telefonbuch eingetragen ist. Florian hat sehr viele Kontakte in seinem Handy eingespeichert und das Telefonbuch beginnt schon zu Staub zu verfallen. Hilf Florian dabei, für jede Nummer in seinem Handy zu überprüfen, ob diese Nummer auch in dem Telefonbuch steht.

Implementiere zwei Funktionen:

```
void init(vector<int> numbers);  
bool query(int number);
```

Zu Beginn wird `init` genau einmal mit dem Inhalt des Telefonbuches aufgerufen. Dieses enthält N Telefonnummern. Dann folgen Q Aufrufe der Funktion `query`, die jeweils `true` zurückgeben soll, wenn die angegebene Nummer im Telefonbuch enthalten ist.

Eingabe

Der Beispiel-Grader liest folgendes Input-Format: Zunächst die Anzahl N von Telefonnummern, dann die einzelnen Telefonnummern im Telefonbuch. Danach die Anzahl Q an Anfragen, und dann die einzelnen zu suchenden Telefonnummern. Die einzelnen Zahlen sind jeweils durch Whitespace (Leerzeichen oder Newlines) getrennt.

Beispiele

Eingabe	Ausgabe
5	1: found
42 1 16 54 100	42: found
3	43: not found
1 42 43	

Subtasks

Allgemein gilt:

- $N \leq 10^6$
- Die Telefonnummern sind Integer zwischen 1 und 10^9



Subtask 1 (30 Punkte): $Q \leq 1000$

Subtask 2 (70 Punkte): $Q \leq 1000000$

Limits

Zeitlimit: 1 s

Speicherlimit: 256 MB