Description

The data about bank transactions consists of a sequence of transactions: the information of each transaction has the following format:

<from_account> <to_account> <money> <time_point> <atm>

In which:

- <from_account>: the account from which money is transferred (which is a string of length from 6 to 20)

- <to_account>: the account which receives money in the transaction (which is a string of length from 6 to 20)

- <money>: amount of money transferred in the transaction (which is an integer from 1 to 10000)

- <time_point>: the time point at which the transaction is performed, it is a string under the format HH:MM:SS (hour: minute: second)

- <atm>: the code of the ATM where the transaction is taken (a string of length from 3 to 10)

Example: T00112233445 T001234002 2000 08:36:25 BIDV (at the ATM BIDV, account T00112233445 transfers 2000$ to account T001234002 at time point 08:36:25 (08 hour, 36 minutes, 25 seconds)

A *transaction cycle* of length *k* starting from account *a*1 is defined to be a sequence of distinct account *a*1, *a*2, …, *ak* in which there are transactions from account *a*1 to *a*2, from *a*2 to *a*3, …, from *ak* to *a*1.{

Write a program that process the following queries:

- ?number_transactions: compute the total number of transactions of the data
- ?total_money_transaction: compute the total amount of money of transactions
- ?list_sorted_accounts: compute the sequence of bank accounts (including sending and receiving accounts) appearing in the transaction (sorted in an increasing (alphabetical) order)
- ?total_money_transaction_from <account>: compute the total amount of money transferred from the account <account>
- ?inspect_cycle <account> k : return 1 if there is a *transaction cycle* of length k, starting from <account>, and return 0, otherwise

Input (stdin)

The input consists of 2 blocks of information: the data block and the query block

- The data block consists of lines:

  - Each line contains the information about a transaction described above
  - The data is terminated by a line containing #
- The query block consists of lines:
  - Each line is a query described above
  - The query block is terminated by a line containing #

Output (stdout)

- Print to stdout (in each line) the result of each query described above

Example

Input

```
T000010010 T000010020 1000 10:20:30 ATM1
T000010010 T000010030 2000 10:02:30 ATM2
T000010010 T000010040 1500 09:23:30 ATM1
T000010020 T000010030 3000 08:20:31 ATM1
T000010030 T000010010 4000 12:40:00 ATM2
T000010040 T000010010 2000 10:30:00 ATM1
T000010020 T000010040 3000 08:20:31 ATM1
T000010040 T000010030 2000 11:30:00 ATM1
T000010040 T000010030 1000 18:30:00 ATM1
#
?number_transactions
?total_money_transaction
?list_sorted_accounts
?total_money_transaction_from T000010010
?inspect_cycle T000010010 3
#
```

Output

9

19500

T000010010 T000010020 T000010030 T000010040

4500

1

Description

Given a rectangle (segments are parallel with the vertical and horizontal axis) specified by two opposite points (x1,y1), (x2,y2) and a circle specified by the center (x,y) and the radius r. Write a program to check if these two shapes intersect with each other (have common points).

Input

- Line 1: contains a positive integer T (1 <= T <= 100000) which is the number of test-cases.
- Subsequent lines are test-cases, each test-case consists of two lines:
  - Line 1: 4 integers x1, y1, x2, y2 (-100000 <= x1, y1, x2, y2 <= 100000)
  - Line 2: 3 integers x, y, r (-100000 <= x, y, r <= 1000000)

Output

- Line i (i = 1,..., T) contains 1 if the answer of the ith test-case is YES, and contains 0, otherwise

Example

Input

2

0 0 4 2

2 3 2

0 0 3 3

10 10 1

Output

```
1

0
```