

Hadeeehhhhh



Description :

Whelp, the prince needs to call someone in their contact to help them buy medicine for their dizziness. Unfortunately for them, the contact is unorganized and all over the place! They can't think too hard so you have to sort them out! Weirdly, they want it to be sorted out through their number????? Weird.....

Constraints:

- $3 < n \leq 100$
- $1 < \text{len}(s) \leq 100$
- $1 < \text{len}(q) \leq 100$
- $0 < c \leq n/2$

Input:

- First input is n , which is the number of contacts.
- After which n contacts will be inputted into the program. Each input line must contain s , which is the name of the contact and q , which is the phone number of the contact.

- Next input is `c`, which is the number of connections (named as close friend) that want to be created. for `c` times the program will ask for two `s`, which is used to find contacts to be connected through a pointer.

Connection that is being made goes both ways ,when you input contact A and B to be connected, then A will have B's contact and B will have A's contact. Contohnya seperti dibawah ini

`a->b`

`b->a`

- The structure of the contact should look like this:

```
typedef struct contact {
    char name[100];
    char phoneNumber[100];
    struct contact *closeFriend;
} contact;
```

Output:

- The output of the program should be the sorted contacts, which is based on the total value of the numbers in `q`.
- The output should contain the index of the contact, then `s` and `q`, separated with dashes “-----”
- **Sort the value ascendingly (Smallest to biggest) USING BUBBLE SORT**

SORTING CRITERIA

- During comparison check if both total value of numbers in `q` from both contacts being compared are similar, then compare the close friend's total value of `q if exists`.
- If **either** of the contact's close friends **doesn't exist**, or if the result is still similar, then skip the sorting step with that pair.

Examples :

- Example 1

Input

```
10
EthanWalker 083174928301
OliviaCarter 083298411752
MasonBrooks 083512974632
AvaHarrison 083782146590
LiamBennett 083401792538
SophiaMitchell 083659214873
NoahDaniels 083918472105
IsabellaFoster 083204857193
JamesSullivan 083719402865
MiaPorter 083574928610
```

2

```
EthanWalker OliviaCarter
MasonBrooks AvaHarrison
```

Output

Contact #1

```
Name : EthanWalker
Phone Number : 083174928301
```

Contact #2

```
Name : NoahDaniels
Phone Number : 083918472105
```

Contact #3

```
Name : OliviaCarter
Phone Number : 083298411752
```

Contact #4

```
Name : MasonBrooks
Phone Number : 083512974632
```

Contact #5

```
Name : LiamBennett
Phone Number : 083401792538
```

Contact #6

Name : IsabellaFoster
Phone Number : 083204857193

Contact #7

Name : AvaHarrison
Phone Number : 083782146590

Contact #8

Name : JamesSullivan
Phone Number : 083719402865

Contact #9

Name : MiaPorter
Phone Number : 083574928610

Contact #10

Name : SophiaMitchell
Phone Number : 083659214873

- **Explanation 1**

The first line specifies 10 contacts to be inserted, which subsequent lines have shown. Then the next line specifies 2 connections with the name of the contact. The comparison for sorting is based on the total number in the phone number (Sophia's total value in the phone number is 56, while Mia's total value is 53)

- **Example 2**

Input

4
LincolnEllis 083748615920
HazelBarnes 083820974156

ConnorFranklin 083950218764

LucyScott 083681940572

0

Contact #1

Contact #1

Name : LincolnEllis

Phone Number : 083748615920

Contact #2

Name : HazelBarnes

Phone Number : 083820974156

Contact #3

Name : ConnorFranklin

Phone Number : 083950218764

Contact #4

Name : LucyScott

Phone Number : 083681940572

- **Explanation 2**

This contact is the same as the previous example, but since there is no connection, sort it as usual.**Hints :**

- Contact name doesn't contain spaces, no need for special delimiter.

Hadeeehhhhh



Description :

Yah, pangeran perlu menghubungi seseorang di daftar kontaknya untuk membantu membeli obat untuk pusingnya. Sayangnya, daftar kontaknya berantakan dan tidak teratur! Mereka tidak bisa berpikir terlalu keras, jadi kamu harus membantunya! Anehnya, mereka ingin hal itu diselesaikan melalui nomor telepon mereka????? Aneh.....

Constraints:

- $3 < n \leq 100$
- $1 < \text{len}(s) \leq 100$
- $1 < \text{len}(q) \leq 100$
- $0 < c \leq n/2$

Input:

- Masukan pertama adalah n , yang merupakan jumlah kontak.
- Setelah itu, n kontak akan dimasukkan ke dalam program. Setiap baris masukan harus berisi s , yang merupakan nama kontak, dan q , yang merupakan nomor telepon kontak.

- Masukan berikutnya adalah c, yang merupakan jumlah koneksi (dinamakan sebagai teman dekat) yang ingin dibuat. Untuk c kali, program akan meminta dua s, yang merupakan nama kontak yang akan dihubungkan melalui penunjuk.

Koneksi yang dibuat bersifat dua arah, saat dibuatnya koneksi A dan B, Kontak A akan punya kontak B dan Kontak B akan punya kontak A will have A's contact. Contohnya seperti dibawah ini.

a->b
b->a

- Struktur kontak harus terlihat seperti ini:

```
typedef struct contact {  
    char name[100];  
    char phoneNumber[100];  
    struct contact *closeFriend;  
} contact;
```

Output:

- Output program harus berupa daftar kontak yang diurutkan berdasarkan nilai total angka dalam q.
- Output harus berisi indeks kontak, diikuti oleh s dan q, dipisahkan dengan tanda hubung "-----"
- **Urutkan nilai secara ascending (dari terkecil ke terbesar)
Menggunakan Bubble Sort**

KRITERIA PENGURUTAN

- Jika selama perbandingan menemui kedua kontak dengan nilai total angka q yang sama, maka bandingkan nilai total teman dekat keduanya jika ada.
- Jika salah satu teman dekat kontak tidak ada atau komparasi total angka dalam q masih sama, maka lewati pengurutan pada kedua kontak tersebut.

Examples :

- Example 1

Input

```
10
EthanWalker 083174928301
OliviaCarter 083298411752
MasonBrooks 083512974632
AvaHarrison 083782146590
LiamBennett 083401792538
SophiaMitchell 083659214873
NoahDaniels 083918472105
IsabellaFoster 083204857193
JamesSullivan 083719402865
MiaPorter 083574928610
```

```
2
```

```
EthanWalker OliviaCarter
MasonBrooks AvaHarrison
```

Output

```
Contact #1
```

```
Name : EthanWalker
Phone Number : 083174928301
```

```
Contact #2
```

```
Name : NoahDaniels
Phone Number : 083918472105
```

```
Contact #3
```

```
Name : OliviaCarter
Phone Number : 083298411752
```

```
Contact #4
```

```
Name : MasonBrooks
Phone Number : 083512974632
```

```
Contact #5
```

Name : LiamBennett
Phone Number : 083401792538

Contact #6

Name : IsabellaFoster
Phone Number : 083204857193

Contact #7

Name : AvaHarrison
Phone Number : 083782146590

Contact #8

Name : JamesSullivan
Phone Number : 083719402865

Contact #9

Name : MiaPorter
Phone Number : 083574928610

Contact #10

Name : SophiaMitchell
Phone Number : 083659214873

- **Explanation 1**

Baris pertama menentukan 10 kontak yang akan dimasukkan, yang ditampilkan pada baris-baris berikutnya. Baris berikutnya menentukan 2 koneksi dengan nama kontak. Perbandingan untuk pengurutan didasarkan pada jumlah total dalam nomor telepon (nilai total Sophia dalam nomor telepon adalah 56, sementara nilai total Mia adalah 53).

- **Example 2**

Input

4
LincolnEllis 083748615920
HazelBarnes 083820974156
ConnorFranklin 083950218764
LucyScott 083681940572
0

Contact #1

Contact #1

Name : LincolnEllis
Phone Number : 083748615920

Contact #2

Name : HazelBarnes
Phone Number : 083820974156

Contact #3

Name : ConnorFranklin
Phone Number : 083950218764

Contact #4

Name : LucyScott
Phone Number : 083681940572

- **Explanation 2**

Kontak yang ini sama dengan contoh sebelumnya, akan tetapi karena tidak ada koneksi, maka urutkan seperti biasanya

Hints :

- Nama kontak tidak mengandung spasi, tidak perlu menggunakan special delimiter.