

[PI035] - Contact-Management

After getting tired of constantly losing contact details of friends and colleagues, you decided to build your own simple contact manager in C. You want the system to handle basic tasks like finding contacts born before a certain year or sorting them alphabetically. You want to start by storing contact information in a structure containing a name, email, date of birth (as an integer in the format YYYYMMDD), and a phone number.

Task

You want to be able to read from the standard input all contacts that you have and want to structure and manage. You are given the number of contacts to be read, an integer n ($1 < n \leq 1000$). Each contact is characterized as follows:

- Name (a single word, up to 100 characters)
- Email (a single word, up to 30 characters)
- Date of birth (as an integer YYYYMMDD, e.g., 19980512)
- Phone (an integer)

After the list of contacts, you are given a flag that defines a different task for you to solve:

- flag=0, then you are also given a month (value from 01 to 12) and want to list all contacts that have birth dates in the same month.
- flag=1, then print the contacts list in alphabetical order of contact names.

Input

The first line of input contains an integer n ($1 \leq n \leq 1000$) — the number of contacts.

The next $n \times 4$ lines contain the contacts data in the order defined previously

Then it follows an integer, the flag. If it is:

- 0 - then an integer m is given with the value of a month ($1 \leq m \leq 12$).
- 1 - nothing else follows.

Output

If flag=0, output all contacts whose birthday is in the same month as the one given in input. Just provide the name and phone, both in same line. **If the contact doesn't exist print NONE.**

If flag=1, we output all contacts sorted alphabetically by the name of each contact.

Given code:

Use this code as a starting point and complete the missing parts. If you decide, you can build your own code from scratch.

```
// Libraries
#include
#include
#include
// Constants
#define MAXCONTACTS 1000
#define MAXNAME 100
#define EOL '\n'
#define SKIPEOL {while (getchar()!='\n') ;}

// new data types
typedef struct {
    // to be implemented
    // define the structure
    // pay attention of functions already given
} Contact;

// Global variables
Contact myContacts[MAXCONTACTS]; // array with my contacts list
```

```

int      nContacts;           // number of contacts in myContacts list

// prototype functions
void printContact(int);
Contact readContact();
void inputMyContacts();
void printContactsWithBirth(int);
char *readName();
void sortContacts();

int main() {
    inputMyContacts();
    int flag;
    scanf("%d", &flag);
    if (flag==0) {
        int month;
        scanf("%d",&month);
        printContactsWithBirth(month); // to be implemented
    } else if (flag==1) { // sort
        sortContacts();           // to be implemented
        for (int i=0; i<nContacts; i++)
            printContact(i);      // to be implemented
    }
}

// print 1 contact
void printContact(int index) {
    // to be implemented
}

// reads a name (line of characters) from standard input
char *readName() {
    char ch, aux[MAXNAME], *name;
    int ctr= 0;
    while ((ch=getchar())!=EOL)
        aux[ctr++]= ch;
    aux[ctr]= '\0';
    name= (char *) malloc(ctr+1);
    strcpy(name, aux);
    return name;
}

// read 1 contact
Contact readContact() {
    Contact c;
    c.name= readName();
    c.email= readName();
    scanf("%d %d",&c.birth, &c.phone);
    SKIPEOL;
    return c;
}

// read a set of contacts
void inputMyContacts() {
    scanf("%d", &nContacts);
    SKIPEOL;
    for (int i= 0; i < nContacts; i++)
        myContacts[i]= readContact();
}

//
void printContactsWithBirth(int month) {
    // to be implemented
}

//
void sortContacts() {
    // to be implemented
}

```

Example 1

Input

```

5
Joao Maria
jmaria@nodomain.pt
19900304
934567899
Maria Joao
mjoao@nodomain.pt
20001112
957634567
Francisco Silva
xicos@nodomain.pt
20030320
967676767
Ana Feliz

```

```
afeliz@nodomain.pt
20051125
987587761
Ricardo Coracao de Leao
rleao@nodomain.pt
20050316
919191911
0
11
```

Output

```
Maria Joao 957634567
Ana Feliz 987587761
```

Example 2

Input

```
5
Joao Maria
jmaria@nodomain.pt
19900304
934567899
Maria Joao
mjoao@nodomain.pt
20001112
957634567
Francisco Silva
xikos@nodomain.pt
20030320
967676767
Ana Feliz
afeliz@nodomain.pt
20051125
987587761
Ricardo Coracao de Leao
rleao@nodomain.pt
20050316
919191911
1
```

Output

```
Ana Feliz
afeliz@nodomain.pt
20051125
987587761
Francisco Silva
xikos@nodomain.pt
20030320
967676767
Joao Maria
jmaria@nodomain.pt
19900304
934567899
Maria Joao
mjoao@nodomain.pt
20001112
957634567
Ricardo Coracao de Leao
rleao@nodomain.pt
20050316
919191911
```