

This file contains all problem descriptions. This exam consists of 4 problems that you need to solve. Problem description for both English and Indonesian are provided in this file. Good luck, have fun!

Berkas ini mengandung semua deskripsi soal. Ujian ini tersusun atas 4 soal yang perlu Anda selesaikan. Deskripsi soal dalam bahasa Inggris dan Indonesia tersedia dalam berkas ini. Selamat mengerjakan!

A. BMI Sorting Machine[EN]

You are given N data where each data consists of ID, name, and height in foot and inch. Then, there are M queries. For each query, given a variable x which represents a height in cm. You are asked for help to display all data whose height is less than or equal to x , ordered descending by the height. If there are 2 data which has the same height, sort ascending by the ID. If no data satisfies the query, output "-1".

Note that 1 foot = 12 inch, and 1 inch = 2,54 cm.

Input Format

The first line consists of an integer N .

The next N lines consist of the given data in the format "A,B,C D" where A, B, C, and D represent ID, name, height in feet, and height in inch, respectively.

The next line consists of an integer M .

The next M lines consist of an integer x which represents the query.

Output Format

Each query is started by "Q", followed by the query number started from 1, followed by ":".

For each query, if there is data to display, then output the data sorted descending based on the height with the following format: "<ID> <Name> <Height in cm>".

If there is no data to display, output "-1".

Constraints

- $1 \leq N \times M \leq 5 \times 10^4$
- ID is an integer between 1 and N . It is guaranteed that 2 data will not have the same ID.
- Length of each name is between 1 and 20 (inclusive). Names consist of Latin alphabet or whitespace. It is guaranteed that the first and last character are not whitespaces.
- Height in foot is between 0 and 7 (inclusive). Height in inch is between 0 and 11 (inclusive).
- $1 \leq x \leq 200$

Subtask 1 (80 points):

- $1 \leq N \leq 10^2$

Subtask 2 (20 points):

- No additional constraint

Sample Input	Sample Output
5 1,John Doe,4 6 2,Anonymous,5 4 3,Vohu Manah,6 1 4,Dora Hariwini,5 4 5,Kyo,6 7 3 165 130 200	Q1: 2 Anonymous 162.56 4 Dora Hariwini 162.56 1 John Doe 137.16 Q2: -1 Q3: 3 Vohu Manah 185.42 2 Anonymous 162.56 4 Dora Hariwini 162.56 1 John Doe 137.16

Notes

In this problem, there are subtasks. If your solution can only solve subtask 1, then you will get **80 points**. For example, your program is not efficient enough such that it can solve the problem with the constraint program $1 \leq N \leq 10^2$ under the 1 second time limit, but unable to solve larger value of N , such as $N = 5 \times 10^4$.

Note that for this problem, if your program can solve subtask 2, then the program can solve subtask 1 as well. Therefore, if your program can solve this problem within all of the constraints, then your solution is considered correct and you will get **100 points**.

B. Record Finder[EN]

You are given N customer data where each data contains AccNumber, name, age, and ballance. The given data is sorted lexicographically ascending based on the AccNumber. Then, there are M queries. For each query, given a variable x . You are asked to tell whether a customer whose AccNumber is x exists. If so, output the customer's data. If the customer does not exist, output "-1".

Input Format

The first line consists of an integer N .

The next N lines contain the given data. Each data consists of AccNumber, name, age, and balance, respectively, separated by commas.

The next line consists of an integer M .

The next M lines consist of an integer x , which represent each query.

Output Format

For each query, if the customer exists, then print the data with the following format: "<Name> <Age> <Balance with 2 decimal places>".

If there is no data to display, output "-1".

Constraints

- $1 \leq N, M \leq 5 \times 10^4$
- All given AccNumber and x in each query are a string consists of exactly 4 upper case Latin alphabet characters. It is guaranteed that there are no 2 data with the same AccNumber.
- Length of the name is between 1 and 20 (inclusive). Names consist of Latin alphabets and whitespaces. It is guaranteed that the first and last character are not whitespaces.
- Ages are between 10 and 100 (inclusive)
- Balances are between 0 and 10^9 . It is guaranteed that the balances are given in 2 decimal places.

Subtask 1 (80 points):

- $1 \leq N, M \leq 10^3$

Subtasks 2 (20 points):

- No additional constraint

Sample Input	Sample Output
--------------	---------------

5 ANGE,Andrew Garfield,18,110933.25 HAIS,Spectra Vgeist,18,25000.00 NERA,Frankie,16,999999999.99 TION,Dorothy White,22,0.00 VVVV,Michael J,17,100000.00 3 NERA JOLY VVVV	Frankie 16 999999999.99 -1 Michael J 17 100000.00
---	---

Notes

In this problem, there are subtasks. If your solution can only solve subtask 1, then you will get **80 points**. For example, your program is not efficient enough such that it can solve the problem with the constraint program $1 \leq N, M \leq 10^3$ under the 1 second time limit, but unable to solve larger value of N , such as $N = M = 5 \times 10^4$.

Note that for this problem, if your program can solve subtask 2, then the program can solve subtask 1 as well. Therefore, if your program can solve this problem within all of the constraints, then your solution is considered correct and you will get **100 points**.

C. Financial Aid During Pandemic[EN]

Prime Minister has announced that the government will give financial aid to reduce the burden of the people during the pandemic. Upon this initiation, recipient will receive money with the following amounts:

- RM1000 for household category B40
- RM600 for household category M40

In this problem, N data are given which contains the category, name, and the amount of money received. You are asked to display all data in a category, B40 or M40. Output the information of the recipients in that category, how many recipients are in that category, and the total amount of money for all of the recipients in that category.

Input Format

Input will be given in a file named **"testdata.in"**.

The first line consists of an integer N .

The next N lines consist of the given data. Each data consists of category, name, and the amount of money received, respectively, separated by commas.

The next line contains the category you want to display, B40 or M40.

Output Format

First, for all recipients under the asked category, output the category, name, and the amount of money received. Print in the order from the input.

In the next line, output "Total Recipients: ", followed by the number of recipients under the asked category.

In the next line, output "Total Amount: ", followed by the total amount of money for all recipients under the category.

Constraints

- $1 \leq N \leq 50$
- Length of name is between 1 and 20 (inclusive). Names consist of upper case Latin alphabet.
- Category can only be B40 or M40.
- It is guaranteed that the amount of money received is in accordance with the category and the provisions that have been explained in the description above.

Sample Input	Sample Output
6 B40, Sofea, 1000 M40, Mikael, 600 B40, Adam, 1000 B40, Irene, 1000 M40, Cyntia, 600 B40, Alif, 1000 B40	B40 Sofea 1000 B40 Adam 1000 B40 Irene 1000 B40 Alif 1000 Total Recipients: 4 Total Amount: 4000
6 B40, Sofea, 1000 M40, Mikael, 600 B40, Adam, 1000 B40, Irene, 1000 M40, Cyntia, 600 B40, Alif, 1000 M40	M40 Mikael 600 M40 Cyntia 600 Total Recipients: 2 Total Amount: 1200

D. Booklet Printing[EN]

When printing a document, usually the first page is printed first, then the second, third, and so on until the last page. However, when printing booklets (folding books), the printing order must be changed so that after folding the paper, the booklet will have correct page order.

In printing booklets, each paper is used to print 4 pages: 2 pages on the front side and 2 pages on the back side. The following steps will show you how to make a booklet so that the order is in the same order as a normal book:

1. Sort the papers according to the correct order.
2. For all papers, make sure that the front side is pointing towards the reader's eye.
3. Fold the papers horizontally downward, so that half of the left sheets will be below the right sheets.
4. Keep the shape of the book by stapling or binding it.

If all these papers are stacked in the right order and the front side of the first sheet is pointing towards the reader's eyes, then it can be guaranteed that the booklet will have the correct page order as the normal book.

For example, a 4-page booklet requires 1 sheet of paper: the front contains the 4th and 1st pages, then the back contains the 2nd and 3rd pages. When the papers are folded, the 1st page becomes the front cover of the book, then the 2nd and 3rd pages are inside, and finally the 4th page becomes the back cover.

Front

4	1
---	---

Back

2	3
---	---

It is known that the number of pages you want to print as a booklet is N . Your task is to create a program that can produce a print order with the minimum possible number of sheets.

Input Format

The first line consists of an integer T , which represents the number of test cases.

For each test case, there is an integer N which represents the number of pages to be printed.

Output Format

For each test case, start with the format "Case #X:" where X is the test case number started from 1.

The next few lines contain the requested booklet printing order, in the format "Sheet P, Q: R, S" where P is the sheet number, Q is the paper part, R and S are the page numbers printed on the paper on the left right in a row.

- For P, the sheet number starts at 1.
- For Q, print "front" if it is the front of the paper, and print "back" if it is the back of the paper.
- For R and S, page numbers start from 1. If the page has run out and there are still sheets left, print "Blank".
- Sort out from the smallest sheet number of papers. For each sheet of paper, remove the front first, then the back.

Read the sample output for more details.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 100$

Sample Input	Sample Output
3 4 14 1	Case #1: Sheet 1, front: 4, 1 Sheet 1, back: 2, 3 Case #2: Sheet 1, front: Blank, 1 Sheet 1, back: 2, Blank Sheet 2, front: 14, 3 Sheet 2, back: 4, 13 Sheet 3, front: 12, 5 Sheet 3, back: 6, 11 Sheet 4, front: 10, 7 Sheet 4, back: 8, 9 Case #3: Sheet 1, front: Blank, 1 Sheet 1, back: Blank, Blank

Notes

The first test case is an example described in the description.

For the second and third test cases, note that the back cover does not contain a page, so print "Blank".

A. BMI Sorting Machine[ID]

Anda diberikan N buah data yang mana setiap data berisi informasi ID, nama, dan tinggi badan dalam kaki dan inci. Kemudian, terdapat M buah pertanyaan. Untuk setiap pertanyaan, diberikan sebuah variabel x yang merepresentasikan tinggi badan dalam cm. Anda diminta untuk menampilkan data yang tinggi badannya kurang dari atau sama dengan x , secara berurutan berdasarkan tinggi secara menurun. Apabila terdapat 2 data yang memiliki tinggi yang sama, urutkan secara menaik berdasarkan ID. Jika tidak ada data yang memenuhi, keluarkan "-1".

Perlu diperhatikan bahwa 1 kaki = 12 inci, dan 1 inci = 2,54 cm.

Format Masukan

Baris pertama tersusun atas sebuah bilangan bulat N .

N baris berikutnya berisi data yang diberikan dengan format "A,B,C D" dimana A, B, C, dan D secara berturut-turut merepresentasikan ID, nama, tinggi badan dalam kaki, dan tinggi badan dalam inci.

Baris berikutnya berisi sebuah bilangan bulat M .

M baris berikutnya berisi sebuah bilangan x yang merepresentasikan setiap pertanyaan.

Format Keluaran

Setiap pertanyaan diawali dengan "Q", dilanjutkan dengan nomor pertanyaan yang dimulai dari 1, dilanjutkan dengan ":".

Untuk setiap pertanyaan, jika ada data yang ditampilkan, maka keluarkan semua data yang diurutkan berdasarkan tinggi secara menurun dengan format sebagai berikut: "<ID> <Nama> <Tinggi dalam cm>".

Jika tidak ada data yang ditampilkan, keluarkan "-1".

Batasan

- $1 \leq N \times M \leq 5 \times 10^4$
- ID adalah bilangan bulat di antara 1 dan N . Dijamin tidak ada 2 data yang memiliki ID yang sama.
- Panjang nama di antara 1 dan 20 (inklusif). Nama tersusun atas karakter alfabet latin dan spasi. Dijamin karakter pertama dan karakter akhir bukan sebuah spasi.
- Tinggi badan dalam kaki di antara 0 dan 7 (inklusif). Tinggi badan dalam inci di antara 0 dan 11 (inklusif).
- $1 \leq x \leq 200$

Subsoal 1 (80 poin):

- $1 \leq N \leq 10^2$

Subsoal 2 (20 poin):

- Tidak ada batasan tambahan

Contoh Masukan	Contoh Keluaran
5 1,John Doe,4 6 2,Anonymous,5 4 3,Vohu Manah,6 1 4,Dora Hariwini,5 4 5,Kyo,6 7 3 165 130 200	Q1: 2 Anonymous 162.56 4 Dora Hariwini 162.56 1 John Doe 137.16 Q2: -1 Q3: 3 Vohu Manah 185.42 2 Anonymous 162.56 4 Dora Hariwini 162.56 1 John Doe 137.16

Keterangan

Pada soal ini terdapat subsoal. Apabila solusi Anda hanya dapat menyelesaikan subsoal 1 saja, maka Anda mendapatkan **80 poin**. Sebagai contoh, program Anda kurang efisien sehingga bisa menyelesaikan soal dengan batasan $1 \leq N \leq 10^2$ dalam batasan waktu 1 detik, namun tidak bisa untuk nilai N yang lebih besar, seperti $N = 5 \times 10^4$.

Perhatikan pula untuk soal ini, apabila program Anda dapat menyelesaikan subsoal 2, maka program tersebut juga dapat menyelesaikan subsoal 1. Oleh karena itu, jika Anda dapat menyelesaikan soal ini dengan semua batasan yang diberikan, maka solusi Anda dianggap benar, dan Anda mendapatkan **100 poin**.

B. Record Finder [ID]

Anda diberikan N buah data nasabah yang mana setiap datanya berisi informasi AccNumber, nama, umur, dan saldo. Data tersebut sudah diurutkan secara leksikografis menaik berdasarkan AccNumber. Kemudian terdapat M buah pertanyaan. Untuk setiap pertanyaan, diberikan sebuah variabel x . Anda diminta untuk memberitahukan apakah ada nasabah yang AccNumber-nya adalah x . Jika ada, cetak semua data dari nasabah tersebut. Jika nasabah tersebut tidak ada, maka keluarkan "-1".

Format Masukan

Baris pertama tersusun atas sebuah bilangan bulat N .

N baris berikutnya berisi data yang diberikan. Setiap datanya tersusun atas AccNumber, nama, umur, dan saldo yang dipisahkan dengan koma.

Baris berikutnya berisi sebuah bilangan bulat M .

M baris berikutnya berisi sebuah bilangan x yang merepresentasikan setiap pertanyaan.

Format Keluaran

Setiap pertanyaan diawali dengan "Query #X:" dimana X adalah nomor pertanyaan yang dimulai dari 1.

Untuk setiap pertanyaan, jika ada nasabah yang ditemukan, maka keluarkan dengan format sebagai berikut: "<Nama> <Umur> <Saldo hingga 2 digit di belakang koma>".

Jika tidak ada data yang ditampilkan, keluarkan "-1".

Batasan

- $1 \leq N, M \leq 5 \times 10^4$
- Semua AccNumber yang diberikan dan x pada masing-masing pertanyaan merupakan sebuah string dengan panjang tepat 4 karakter yang tersusun atas huruf besar latin alfabet. Dijamin tidak ada 2 data yang memiliki AccNumber yang sama.
- Panjang nama di antara 1 dan 20 (inklusif). Nama tersusun atas karakter alfabet latin dan spasi. Dijamin karakter pertama dan karakter akhir bukan sebuah spasi.
- Umur di antara 10 dan 100 (inklusif)
- Saldo di antara 0 dan 10^9 . Dijamin saldo akan diberikan dalam 2 digit di belakang koma.

Subsoal 1 (80 poin):

- $1 \leq N, M \leq 10^3$

Subsoal 2 (20 poin):

- Tidak ada batasan tambahan

Contoh Masukan	Contoh Keluaran
5 ANGE,Andrew Garfield,18,110933.25 HAIS,Spectra Vgeist,18,25000.00 NERA,Frankie,16,999999999.99 TION,Dorothy White,22,0.00 VVVV,Michael J,17,100000.00 3 NERA JOLY VVVV	Frankie 16 999999999.99 -1 Michael J 17 100000.00

Keterangan

Pada soal ini terdapat subsoal. Apabila solusi Anda hanya dapat menyelesaikan subsoal 1 saja, maka Anda mendapatkan **80 poin**. Sebagai contoh, program Anda kurang efisien sehingga bisa menyelesaikan soal dengan batasan $1 \leq N, M \leq 10^3$ dalam batasan waktu 1 detik, namun tidak bisa untuk nilai N dan M yang lebih besar, seperti $N = M = 5 \times 10^4$.

Perhatikan pula untuk soal ini, apabila program Anda dapat menyelesaikan subsoal 2, maka program tersebut juga dapat menyelesaikan subsoal 1. Oleh karena itu, jika Anda dapat menyelesaikan soal ini dengan semua batasan yang diberikan, maka solusi Anda dianggap benar, dan Anda mendapatkan **100 poin**.

C. Financial Aid During Pandemic[ID]

Perdana Menteri telah mengumumkan bahwa pemerintah akan memberikan bantuan sosial untuk mengurangi beban rakyat selama pandemi. Atas inisiasi ini, penerima akan mendapat dana dengan jumlah sebagai berikut:

- RM1000 untuk rumah tangga kategori B40
- RM600 untuk rumah tangga kategori M40

Pada soal ini, N buah data yang berisi informasi tentang kategori, nama, dan dana bantuan sosial yang diterima. Anda diminta untuk menampilkan data-data untuk suatu kategori, B40 atau M40. Keluarkan informasi dari penerima dengan kategori tersebut, banyaknya penerima dari kategori tersebut, dan total dana dari semua penerima dalam kategori tersebut.

Format Masukan

Input akan diberikan melalui suatu file dengan nama `"testdata.in"`.

Baris pertama tersusun atas sebuah bilangan bulat N .

N baris berikutnya berisi data yang diberikan. Setiap datanya tersusun atas kategori, nama, dan dana yang diterima, masing-masing dipisahkan dengan sebuah koma.

Baris berikutnya berisi kategori yang ingin ditampilkan, B40 atau M40.

Format Keluaran

Pertama, untuk semua penerima yang berada pada kategori yang ditanyakan, keluarkan kategori, nama, dan dana yang diterima. Tampilkan sesuai dengan urutan pada masukan.

Pada baris berikutnya, keluarkan `"Total Recipients: "`, yang dilanjutkan dengan banyaknya penerima pada kategori yang ditanyakan.

Pada baris berikutnya, keluarkan `"Total Amount: "`, yang dilanjutkan dengan total dana dari semua penerima dalam kategori ini.

Batasan

- $1 \leq N \leq 50$
- Panjang nama di antara 1 dan 20 (inklusif). Nama tersusun atas karakter alfabet latin.
- Kategori yang diberikan hanya dapat berupa B40 atau M40.
- Dijamin jumlah dana yang diterima sudah sesuai dengan kategorinya sesuai dengan ketentuan yang sudah dijelaskan pada deskripsi soal.

Contoh Masukan	Contoh Keluaran
6 B40, Sofea, 1000 M40, Mikael, 600 B40, Adam, 1000 B40, Irene, 1000 M40, Cyntia, 600 B40, Alif, 1000 B40	B40 Sofea 1000 B40 Adam 1000 B40 Irene 1000 B40 Alif 1000 Total Recipients: 4 Total Amount: 4000
6 B40, Sofea, 1000 M40, Mikael, 600 B40, Adam, 1000 B40, Irene, 1000 M40, Cyntia, 600 B40, Alif, 1000 M40	M40 Mikael 600 M40 Cyntia 600 Total Recipients: 2 Total Amount: 1200

D. Booklet Printing[ID]

Pada saat mencetak dokumen, biasanya halaman pertama dicetak pertama, kemudian halaman kedua, ketiga, dan seterusnya hingga halaman terakhir. Akan tetapi, pada saat mencetak booklet (buku lipat), urutan pencetakan harus diubah agar setelah dilipat, hasilnya sesuai.

Dalam mencetak booklet, setiap kertas digunakan untuk mencetak 4 halaman: 2 di depan, dan 2 di belakang. Berikut adalah cara untuk membuat booklet sehingga urutannya menjadi sesuai seperti buku biasa:

1. Urutkan kertas sesuai dengan urutan yang tepat.
2. Untuk semua kertas, arahkan bagian depan arah ke mata pembaca.
3. Lipat tumpukan tersebut secara bersamaan secara horizontal ke bawah, sehingga separuh kiri lembar akan berada di bawah separuh kanan.
4. Jaga bentuk buku tersebut dengan men-staples atau mengikat bagian lipatan buku.

Apabila semua kertas ini ditumpuk dengan urutan yang tepat, jika lembar pertama bagian depan mengarah ke mata pembaca, setelah dilipat, urutan halaman nya benar seperti buku biasa.

Sebagai contoh, sebuah booklet dengan 4 halaman membutuhkan 1 lembar kertas: bagian depan berisi halaman ke-4 dan ke-1, kemudian bagian belakang berisi halaman ke-2 dan ke-3. Apabila dilipat di tengah, maka halaman ke-1 menjadi *cover* depan dari buku tersebut, kemudian halaman ke-2 dan ke-3 berada di dalam, dan terakhir halaman ke-4 menjadi *cover* belakang.

Front

4	1
---	---

Back

2	3
---	---

Diketahui bahwa jumlah halaman yang ingin Anda cetak dalam bentuk booklet adalah N . Tugas Anda adalah membuat sebuah program yang dapat mengeluarkan urutan pencetakannya dengan jumlah lembar kertas seminimum mungkin.

Format Masukan

Baris pertama berisi sebuah bilangan bulat T , yang merepresentasikan banyaknya kasus uji.

Untuk setiap kasus uji, terdapat sebuah bilangan bulat N yang merepresentasikan banyaknya halaman yang ingin dicetak.

Format Keluaran

Untuk setiap kasus uji, awali dengan format "Case #X:" dimana X adalah nomor kasus uji yang dimulai dari 1.

Beberapa baris berikutnya berisi urutan pencetakan booklet yang diminta, dengan format “Sheet P, Q: R, S” dimana P adalah nomor lembar kertas, Q adalah bagian kertas, R dan S adalah nomor halaman yang dicetak pada bagian kertas tersebut di sisi kiri dan kanan secara berturut-turut.

- Untuk P, nomor lembar kertas dimulai dari 1.
- Untuk Q, keluarkan “front” jika merupakan bagian depan kertas, dan keluarkan “back” jika merupakan bagian belakang kertas.
- Untuk R dan S, nomor halaman dimulai dari 1. Apabila halaman sudah habis dan masih ada lembar yang tersisa, keluarkan “Blank”.
- Urutkan keluarkan dari nomor lembar kertas paling kecil. Untuk setiap lembar kertas, keluarkan yang bagian depan terlebih dahulu, baru yang bagian belakang.

Perhatikan contoh keluaran untuk lebih jelasnya.

Batasan

- $1 \leq T \leq 10$
- $1 \leq N \leq 100$

Contoh Masukan	Contoh Keluaran
3 4 14 1	Case #1: Sheet 1, front: 4, 1 Sheet 1, back: 2, 3 Case #2: Sheet 1, front: Blank, 1 Sheet 1, back: 2, Blank Sheet 2, front: 14, 3 Sheet 2, back: 4, 13 Sheet 3, front: 12, 5 Sheet 3, back: 6, 11 Sheet 4, front: 10, 7 Sheet 4, back: 8, 9 Case #3: Sheet 1, front: Blank, 1 Sheet 1, back: Blank, Blank

Penjelasan

Kasus uji pertama merupakan contoh yang dijelaskan pada deskripsi.

Untuk kasus uji kedua dan ketiga, perhatikan bahwa *cover* belakang memang tidak terdapat halaman, sehingga keluarkan “Blank”.