

In this homework, we provide two files, classA.txt and classB.txt

In classA.txt:

- 1. The data in classA.txt is student information in the class, A. Each row is the information about a student.
- 2. There are three columns
 - 2.1. “No” column records id number.
 - 2.2. “Name” column records name.
 - 2.3. “Birthday” column records birthday.
(ex: 2002/1/2 means “year 2002” , “February” and “day 2”).

In classB.txt:

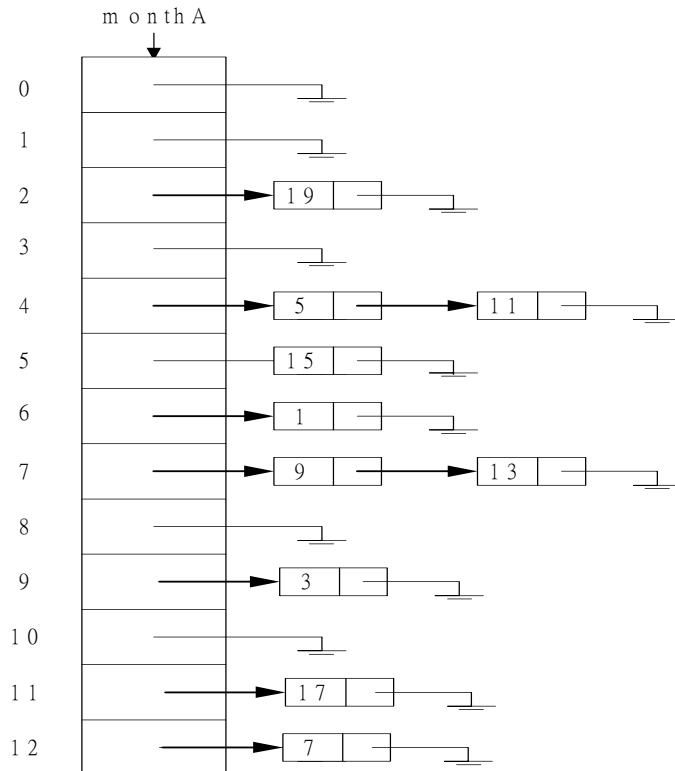
- The way to interpret is the same as classA.txt

Now, use structure definition to define a new type, student. There are three members in the variable of type student. “id” stands for id number for a student. “birthMonth” is the month which a student was born in. As the following code segment shows:

```
struct personalData{
    int id;
    int birthMonth;
    struct personalData *link;
};
```

```
typedef struct personalData student;
```

- a. Use the type we just defined and the linked list structure, please write a program to save the data in classA.txt into a structure as the following:



In the above figure, **monthA** is a one dimension array of type **students***, whose length is 13. In that array:

MonthA[0] = 0 (Don't use this element)

MonthA[1] is the pointer which points to the linked list of students who were born in January.

MonthA[2] is the pointer which points to the linked list of students who were born in February.

.

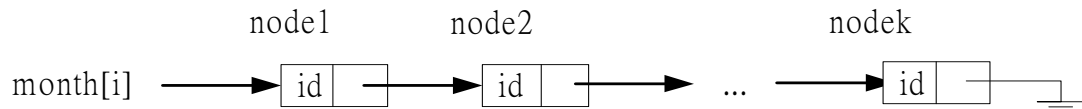
.

.

MonthA[12] is the pointer which points to the linked list of students who were born in December

Besides, every node in linked lists is of type **student**. If no one was born in a month (ex month i), set monthA[i] = 0.

After you save the data of all students into above structure, every node in linked list must be connected in non-decreasing order according to their id number. For example, if a list has k nodes as the following figure (node1 ~ nodek):

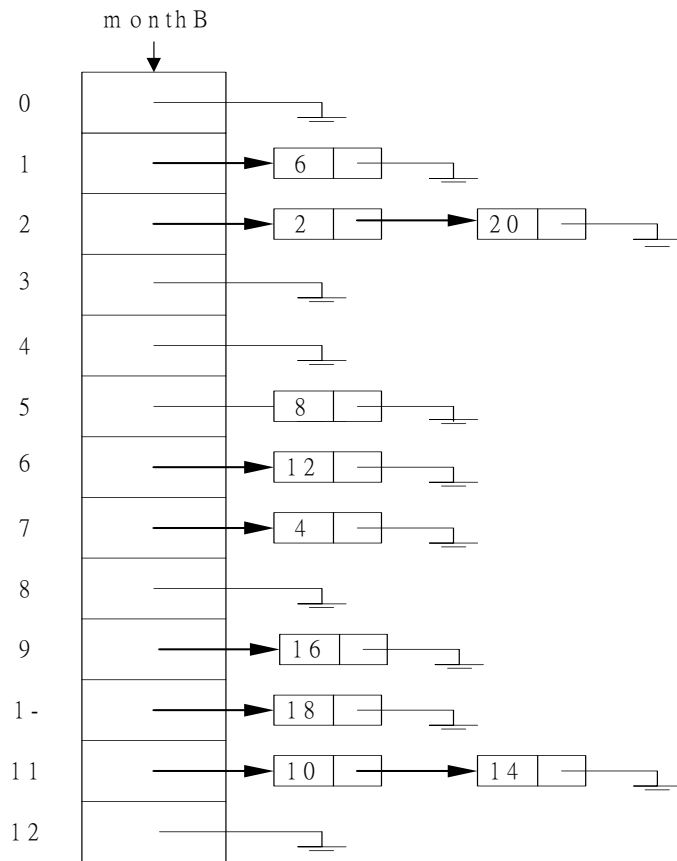


If nodes are connected in non-decreasing order according to their id member, then $\text{node1.id} \leq \text{node2.id} \leq \dots \leq \text{nodek.id}$.

Finally, output the result as the following way:

month	students' id
1	
2	19
3	
4	5,11
5	15
6	1
7	9,13
8	
9	3
10	
11	17
12	7

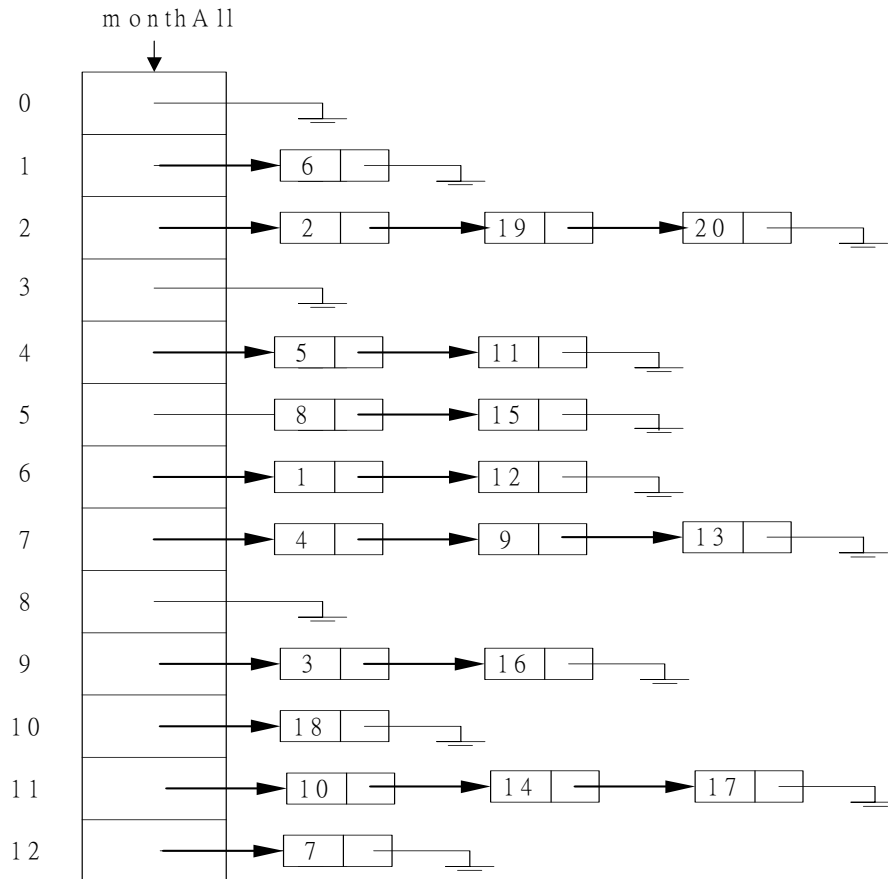
b. Use the same rules to save the data in classB.txt as the following figure shows:



Finally, output the result as part a does.

month	students' id
1	6
2	2,20
3	
4	
5	8
6	12
7	4
8	
9	16
10	18
11	10,14
12	

c. Please write a program to merge the data in **monthA** and **monthB** into the following structure:



And, you must make your program support two output ways, as 3.1 and 3.2 state.

3.1 Users can input an integer. Then, program will output all students who were born in that month according to that integer:

```
Input a month: 2
=====
2,19,20
```

```
Input a month: 3
=====
No student was born in this month!
```

3.2 Use information saved in program to output the data to the file, result.txt, as the following:

No	birth month
1	6

2	2
3	9
4	7
5	4
6	1
7	12
8	5
9	7
10	11
11	4
12	6
13	7
14	11
15	5
16	9
17	11
18	10
19	2
20	2

where each row records data about a student. “id” field records the student id. “birth month” field records the month that a student was born in.