Exercise 8

1. List, Set, TreeMap

Read File and Print it out.

- Read File and print the data inside with 3 types of container : `TreeSet` , `TreeMap` , `ArrayList`

- The path of file is entered by program argument.

  - Your program can read program arguments with String[] args.

- The file contains many lines of number sequels.

- Each row has a numeric array, and each number is separated by a comma(,)

- For `map` container, the key of value is :

  - `[linenumber:indexnumber]`
  - The type of key MUST BE `String`
  - `linenumber` is the number of line array is written on file.
  - `indexnumber` is the index of value in array.

- Print out data in set or map container with its iterator.

- Your program should print out each line number, and print each numbers in containers

  - When printing numbers in `TreeMap` container, print out line number with it.
    - `KEY==VALUE`

- Output example shows that program works in that way.

- You can change each container into any other container that implements same interface.

Input file example(ex_input.in)

```
1,1,1,1,1,1,1,1,1,1,1
10,9,8
3,2,62,62,3
```

Output example

```
[line 1]
List_container : 1,1,1,1,1,1,1,1,1,1,1
Set_container : 1
Map_container : [1:10]==1,[1:11]==1,[1:1]==1,[1:2]==1,[1:3]==1,[1:4]==1,[1:5]==1,[1:6]==1,[1:7]==1,[1:8]==1,[1:9]==1
[line 2]
List_container : 10,9,8
Set_container : 8,9,10
Map_container : [2:1]==10,[2:2]==9,[2:3]==8
[line 3]
List_container : 3,2,62,62,3
Set_container : 2,3,62
Map_container : [3:1]==3,[3:2]==2,[3:3]==62,[3:4]==62,[3:5]==3
```

2. intensive data modifying

description

- implement a simple interpreter

operations

```
// insert <number>
i <number>

// delete <number>
d <number>

// print 1 if the program contains <number>, otherwise print 0
c <number>

// quit program
q
```

- the program has a data store that can add and delete data
- like the set in math, multiple insertions of same number are treated like a single insert
- deletion of nonexistent numbers should be ignored

input example

```
i 0
i 0
c 0
c 1
i 1
c 1
d 0
c 0
q
```

output example

```
1
0
1
0
```