

Name: SAKSHAM MISHRA

DATE:06/03/2022

Coding Problems for practice

1) Java Arraylist

Input Format

The first line has an integer n . In each of the next n lines there will be an integer d denoting number of integers on that line and then there will be d space-separated integers. In the next line there will be an integer q denoting number of queries. Each query will consist of two integers x and y .

Constraints

$$1 \leq n \leq 20000$$

$$0 \leq d \leq 50000$$

$$1 \leq q \leq 1000$$

$$1 \leq x \leq n$$

Each number will fit in signed integer.

Total number of integers in n lines will not cross 100000.

Output Format

In each line, output the number located in y position of x line. If there is no such position, just print "ERROR!"

Sample Input

```
5
5 41 77 74 22 44
1 12
4 37 34 36 52
0
3 20 22 33
5
1 3
3 4
3 1
4 3
5 5
```

Sample Output

74
52
37
ERROR!
ERROR!

Solution :

```
package converter;

import java.util.ArrayList;
import java.util.Scanner;

public class Str {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();
        ArrayList<ArrayList<Integer> > arr =
            new ArrayList<ArrayList<Integer> >(a);

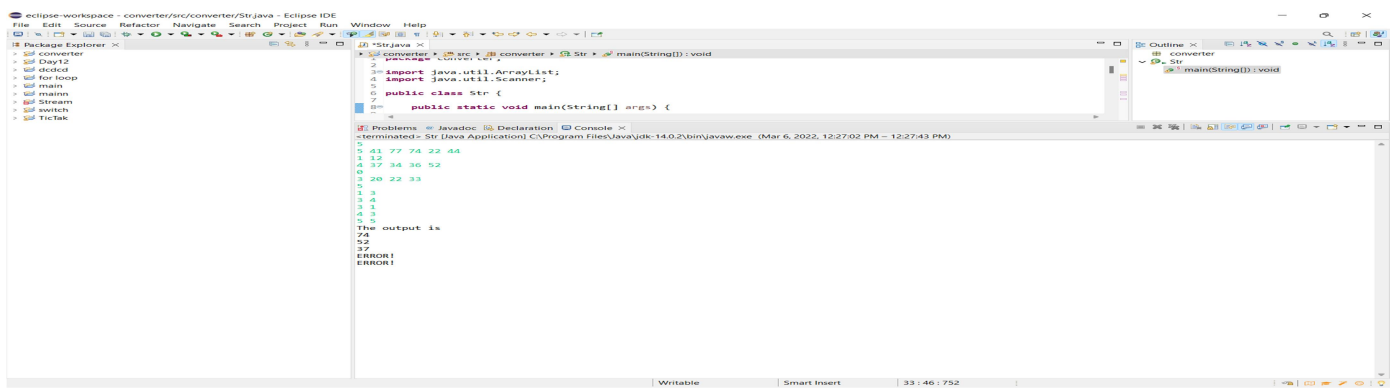
        for(int i=0;i<a;i++) {
            int num = sc.nextInt();
            ArrayList<Integer> a1 = new ArrayList<Integer>();

            for(int k=0;k<num;k++) {
                a1.add(sc.nextInt());
            }
            arr.add(a1);
        }

        System.out.println("The output is");
        int q = sc.nextInt();
        for(int i=0;i<q;i++) {
            int x1 = sc.nextInt()-1;
            int y1 = sc.nextInt()-1;

            if(y1>(arr.get(x1).size()) - 1) {
                System.out.println("ERROR!");
            }

            else {
                System.out.println(arr.get(x1).get(y1));
            }
        }
    }
}
```



2) Java List

Input Format

The first line contains an integer, (the initial number of elements in).

The second line contains space-separated integers describing .

The third line contains an integer, (the number of queries).

The subsequent lines describe the queries, and each query is described over two lines:

- ⑩ If the first line of a query contains the String **Insert**, then the second line contains two space separated integers , and the value must be inserted into at index .
- ⑩ If the first line of a query contains the String **Delete**, then the second line contains index , whose element must be deleted from .

Constraints

- ⑩ $1 \leq N \leq 4000$
- ⑩ $1 \leq Q \leq 4000$
- ⑩ Each element in is a *32-bit integer*.

Output Format

Print the updated list as a single line of space-separated integers.

Sample Input

```
5
12 0 1 78 12
2
Insert
5 23
Delete
```

0

Sample Output

0 1 78 12 23

Solution:

```
package converter;
```

```
import java.util.ArrayList;
```

```
import java.util.Iterator;
```

```
import java.util.List;
```

```
import java.util.Scanner;
```

```
public class Java02 {
```

```
    public static void main(String[] args) {  
        List<Integer> list = new ArrayList<>();
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int n = sc.nextInt();
```

```
        for(int i=0;i<n;i++) {  
            list.add(sc.nextInt());  
        }
```

```
        int q = sc.nextInt();  
        for(int i=1;i<=q;i++) {
```

```
            String c = sc.next();
```

```
            if(c.equals("Insert")) {  
                int a = sc.nextInt();  
                int b = sc.nextInt();  
                list.add(a,b);  
            }
```

```
            else {  
                int z = sc.nextInt();  
                list.remove(z);  
            }
```

```
        }
```

```
        Iterator<Integer> iterate = list.iterator();
```

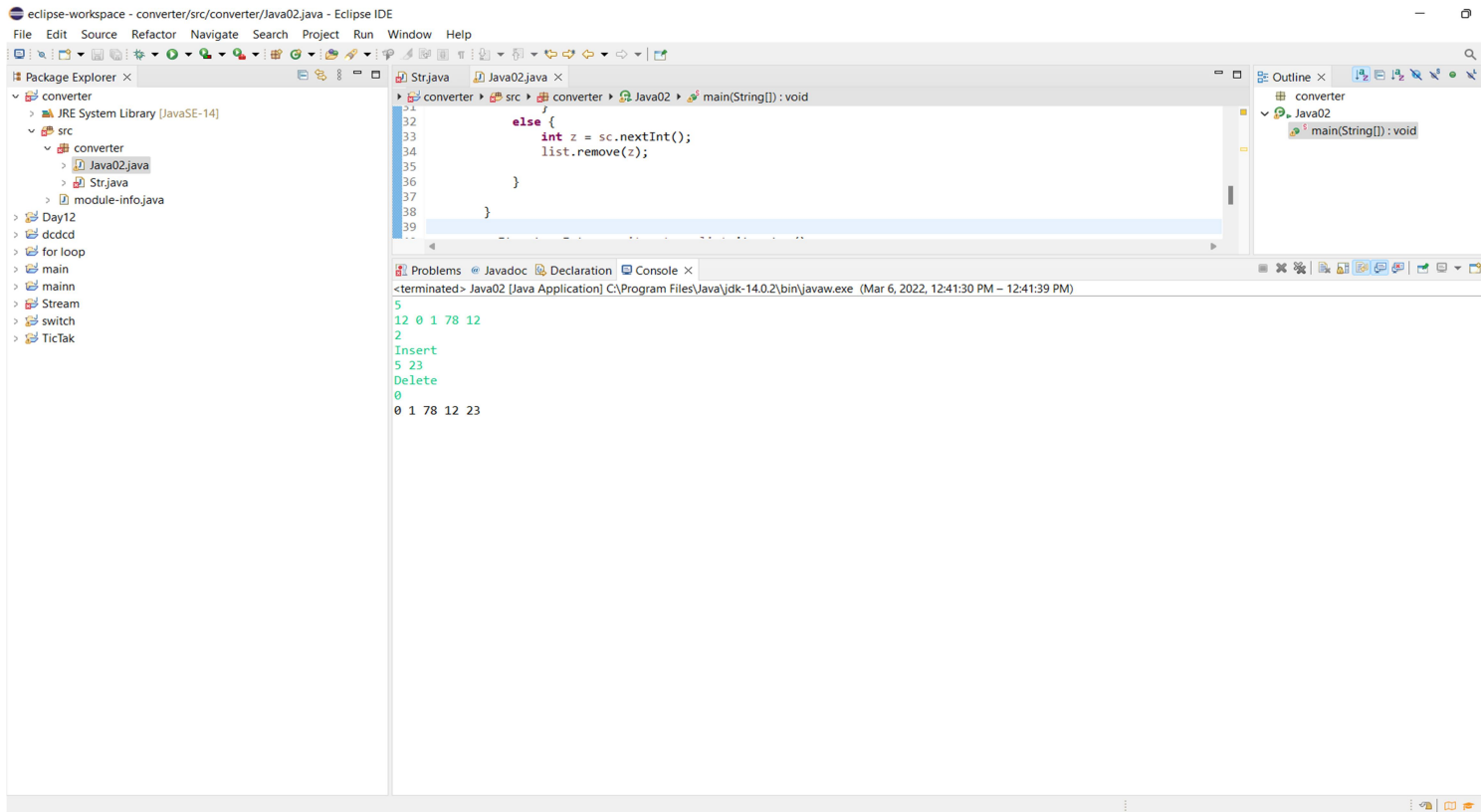
```
        while(iterate.hasNext()) {  
            System.out.print(iterate.next()+" ");  
        }
```

```

    }

}

```



3) Java Map

You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person.

Input Format

The first line will have an integer denoting the number of entries in the phone book. Each entry consists of two lines: a name and the corresponding phone number.

After these, there will be some queries. Each query will contain a person's name. Read the queries until end-of-file.

Constraints:

A person's name consists of only lower-case English letters and it may be in the format 'first-name last-name' or in the format 'first-name'. Each phone number has exactly 8 digits without any leading zeros.

$1 \leq n \leq 100000$

$1 \leq \text{Query} \leq 100000$

Output Format

For each case, print "Not found" if the person has no entry in the phone book. Otherwise, print the person's name and phone number. See sample output for the exact format.

To make the problem easier, we provided a portion of the code in the editor. You can either complete that code or write completely on your own.

Sample Input

```
3
uncle sam
99912222
tom
11122222
harry
12299933
uncle sam
uncle tom
harry
```

Sample Output

```
uncle sam=99912222
Not found
harry=12299933
```

Solution:

```
package converter;

import java.util.HashMap;
import java.util.Scanner;

public class Java_03 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HashMap<String, Integer> hash = new HashMap<>();
        Scanner sc = new Scanner(System.in);
        int a=sc.nextInt();
        sc.nextLine();
        for(int i=0;i<a;i++)
        {
            String name1=sc.nextLine();
            int phone1=sc.nextInt();
            sc.nextLine();
            hash.put(name1,phone1);
        }
        while(sc.hasNext())
        {

            String s=sc.nextLine();
            try
            {
                int out1=hash.get(s);
                System.out.println(s+"="+out1);
            }
            catch(Exception e)
            {
                System.out.println("Not found");
            }
        }
    }
}
```

eclipse-workspace - converter/src/converter/Java_03.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer ×

- converter
 - JRE System Library [JavaSE-14]
 - src
 - converter
 - Java_03.java
 - Java02.java
 - Str.java
 - module-info.java
 - Day12
 - dcdd
 - for loop
 - main
 - mainn
 - Stream
 - switch
 - TicTak

Str.java Java02.java Java_03.java ×

converter > src > converter > Java_03

```
1 package converter;
2
3 import java.util.HashMap;
```

Outline ×

- converter
 - Java_03
 - main(String[]): void

Problems Javadoc Declaration Console ×

Java_03 [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (Mar 6, 2022, 12:54:23 PM)

```
3
uncle sam
99912222
tom
11122222
harry
12299933
uncle sam
uncle tom
harry
uncle sam=99912222
Not found
harry=12299933
```

Writable Smart Insert 37 : 1 : 883