

题目描述：

实现一个支持优先级的队列，高优先级先出队列；同优先级时先进先出。

如果两个输入数据和优先级都相同，则后一个数据不入队列被丢弃。

队列存储的数据内容是一个整数。

输入描述：

一组待存入队列的数据（包含内容和优先级）

输出描述：

队列的数据内容（优先级信息输出时不再体现）

补充说明：

不用考虑输入数据不合法的情况，测试数据不超过 100 个

示例 1

输入：

(10,1),(20,1),(30,2),(40,3)

输出：

40,30,10,20

说明：

输入样例中，向队列写入了 4 个数据，每个数据由数据内容和优先级组成。

输入和输出内容都不含空格。

数据 40 的优先级最高，所以最先输出，其次是 30；10 和 20 优先级相同，所以按输入顺序输出。

示例 2

输入：

(10,1),(10,1),(30,2),(40,3)

输出：

40,30,10

说明：

输入样例中，向队列写入了 4 个数据，每个数据由数据内容和优先级组成。

输入和输出内容都不含空格。

数据 40 的优先级最高，所以最先输出，其次是 30；两个 10 和 10 构成重复数据，被丢弃一个。

```
import java.util.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

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

```
        String str = scanner.nextLine();
```

```
        str = str.replaceAll("\\\\(", "").replace(")", ",").replace(")", "");
```

```
        String[] strArr = str.split(",");
```

```
        PriorityQueue<Integer> priorityQueue = new PriorityQueue<>((a, b) -> a[1] == b[1] ?
```

```
        a[2] - b[2] : b[1] - a[1]);
```

```
        Set<String> set = new HashSet<>();
```

```
        int i = 0;
```

```

        for (String subStr : strArr) {
            if (set.contains(subStr)) {
                continue;
            }
            set.add(subStr);
            String[] subArr = subStr.split(",");
            priorityQueue.add(new Integer[]{Integer.parseInt(subArr[0]),
Integer.parseInt(subArr[1]), i++});
        }

        StringJoiner sj = new StringJoiner(",");
        while (priorityQueue.size() > 0) {
            sj.add(String.valueOf(priorityQueue.poll()[0]));
        }

        System.out.println(sj.toString());
    }
}

```