

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.ArrayDeque;
```

```
import java.util.Deque;
```

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

```
public class Main {
```

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

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

```
        String[] line1 = scanner.nextLine().replaceAll("\\(|\\)", "").split(",");
```

```
        Deque<int[]> array = new ArrayDeque<>();
```

```
        Deque<int[]> cache = new ArrayDeque<>();
```

```
        for (int i = 0; i < line1.length; i++) {
```

```
            int value = Integer.parseInt(line1[i]);
```

```
            int level = Integer.parseInt(line1[i + 1]);
```

```
            if (array.isEmpty()) array.push(new int[]{value, level});
```

```
            else {
```

```
                while (array.size() > 0 && array.peek()[1] < level) {
```

```
                    cache.push(array.pop());
```

```
                }
```

```
                if (array.isEmpty()) array.push(new int[]{value, level});
```

```
                else if (!(array.peek()[1] == level && array.peek()[0] == value)) array.push(new
```

```
int[]{value, level});
```

```
                while (cache.size() > 0) {
```

```
                    array.push(cache.pop());
```

```
                }
```

```
            }
```

```
            i++;
```

```
        }
```

```
        int i = 0;
```

```
        while (array.size() > 0) {
```

```
            if (i != 0) System.out.print(",");
```

```
            System.out.print(array.pollLast()[0]);
```

```
            i++;
```

```
        }
```

```
    }
```

```
}
```