作业： Input: 一个包含（ ）的 String

output: 倒序 input string，但（ ）作为一个整体 内部不被倒序

例· "abc (def) gh " → " hg (def) cba"

练习

```
┌─────────────────────────────────┐
│   Item                           │
├─────────────────────────────────┤
│ + map : Map                      │
├─────────────────────────────────┤
│ + Item ( ) : void                │
│ + Item ( Item[][] map ) : void   │     ※ 没用到
│  { 先用上一个构造器创建一个 Item    │       随机放
│    再将其放入 map 中的一个空位置 }  │
├─────────────────────────────────┤
│ + distance ( Item another ) : int │
│  { 先判断两个 Item 是否在同一个 Map中，若是，则 │
│    计算两者间的距离；若否，则回传 -1 }          │
└─────────────────────────────────┘
```

```
┌─────────────────────────────────┐
│   Map                            │
├─────────────────────────────────┤
│ + map : Item[][]                 │
├─────────────────────────────────┤
│ + Map ( int size )               │
│ { this.map = Item[size][size] }  │
└─────────────────────────────────┘
```

Main 中完成：

Map map = new Map (10)

Item i1 = new Item (map)

Item i2 = new Item (map)

Item i3 = new Item (null)

sout ( i1.distance (i2) )

sout ( i1.distance (i3) )

内容： 1. ArrayList : add(), get(int index), set(int index, Object obj)

　　　　　　 remove(int index) , size() , indexOf ( Object obj)

　　　　　　 contains (Object obj) , clone() , subList ( int from, int to)

　　　　　　　　　　　　　　　　　　　　 左闭右开

　　　　　　 sort ( )

2. HashSet : 同 ArrayList，但不允许重复元素

3. HashMap : 是一个 key - value set : put (key, value) , get( key) , remove (key)

　　 containsKey (key) , containsValue (v) , putIfabsent (··,··) , clone ( ), keySet ( )

　　　　　　　　　　　　　　　　　　　　　　　　　 forEach ( )

※ 作业要求

※ lambda