**num628**

1. java中定义数组格式： int[] arr = new int[]{1,2,3,4,5};
2. 对数组进行排序：Arrays.sort(arr);
3. 两个数比较大小：Math.max(a,b);
4. java中最大正数为： Integer.MAX\_VALUE； 最大负数为：Integer.MIN\_VALUE

**Num118**

1. Java中使用List来表示二维数组格式：List<List<Integer>> list = new ArrayList<>();
2. Java中使用List来表示一维数组格式：List<Integer> col = new ArrayList<>();
3. 向一维数组中添加元素：col.add(1);
4. 向二维数组中添加元素：list.add(col); 也就是先添加一维数组（表示每行元素），再把一维数组添加到另一个一维数组中。
5. 获取二位数组第i行j列的元素：list.get(i).get(j); //这里数组下标以1开始。

**Num69**

1. java中log运算：Math.log(x);
2. java中ex运算：Math.exp(x);

**Num56**

1. 字符串str=”abc”的长度：str.length();
2. 字符串str=”abc”转化为字符数组：char[] c = str.toCharArray();

**Num58**

1. 用特定符号切割字符串：String[] arr = str.split(“ “);

**Num73**

1. List系列集合：添加的元素是有序、可重复、有索引。
2. Set系列集合：添加的元素是无序、不重复、无索引。
3. Set的定义格式：Set<Integer> s = new HashSet<>();
4. 向Set中添加元素：s.add();

**Num94**

1、Java中在返回值为void的方法中使用语句： return; 作用是结束方法的执行。

**Num503**

1. continue：跳出本次循环继续下一次循环。也就是，跳过本次循环体中剩下尚未执行的语句，立即进行下一次的循环条件判定。
2. break：跳出一个循环体，完全结束一个循环。

如果break语句包含在[嵌套循环](https://so.csdn.net/so/search?q=%E5%B5%8C%E5%A5%97%E5%BE%AA%E7%8E%AF&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_48374573/article/details/_blank)里，它只跳出最里面的循环。

**Num387**

1、Java中获取字符串索引为i的字符：char ch = s.charAt(i); 其中s是一个字符串，如”ababba”。

**Num187**

1. Java中HashMap 的getOrDefault()方法获取指定key对应的value，如果找不到key，那么就返回默认值。

**其语法为**：V k = map.getOrDefault(Object key, V defaultValue);

注：map是HashMap类的一个对象。k是默认的返回值，即defaultValue（当指定的key并不存在该映射关系中，返回该默认值defaultValue）。

1. Java中从指定的字符串s中，从位置i开始向后截取长度为k的字串。String substring = s.substring(i, i+k);

注：该方法截取的字串含头不含尾，即包含索引为i的字符，不包含索引为i+k的字符。

**Num1**

1. Java中HashMap的定义：Map<Integer,Integer> map = new HashMap<>();
2. 判断该map集合中是否包含指定的key值：boolean m = map.containKey(key);存在返回true，不存在返回false；
3. 判断该map集合中是否包含指定的value值：boolean m = map.containValue(value);存在返回true，不存在返回false；

**Num316**

1. Java中提高字符串的操作效率，StringBuilder类。

构造方法：

public StringBuilder()：创建一个空白可变字符串对象，不包含任何内容。

public StringBuilder(String str)：根据字符串的内容，来创建可变字符串对象。

成员方法：

public StringBuilder append(任意类型)：添加数据，并且返回对象本身。

public StringBuilder reverse()：反转容器中的内容。

public int length()：返回长度（字符的个数）。

public String toString()：通过toString()就可以实现把StringBuilder转换为String。

例如：

StringBuilder sb = new StringBuilder();

sb.append(char);

sb.reverse().toString();

1. Java中Stack类（后进先出）

定义：Stack<Character> st = new Stack<>();

成员方法：

boolean empty(); 判断堆栈是否为空。

Object peek()：查看堆栈顶部的对象，但不从堆栈中移除。

Object pop()：移除堆栈顶部的对象，并作为此函数的返回值。

Object push(Object element)：把该元素压入栈顶。

int search(Object element)：返回对象在堆栈中的位置，以1为基数。

**Num215**

Java中堆的实现，优先级队列PriorityQueue。

1. 构造方法

空参构造方法：PriorityQueue<Integer> queue = new PriorityQueue<>(); 创建一个空的优先级队列，默认容量为11，默认为升序排序，小根堆。

有参构造方法：PriorityQueue<>(int initialCapacity, new Comparator<Integer>(){...});创建一个初始容量为initialCapacity的优先级队列。

例如：

PriorityQueue<Integer> queue = new PriorityQueue<>(count, new Comparator<Integer>() {  
 @Override  
 public int compare(Integer o1, Integer o2) {  
 //升序排序，小根堆  
 return o1-o2;  
 //降序排序，大根堆  
 return o2-o1;  
 }  
});

1. 成员方法

queue.add(key); 将指定的元素key插入队列。如果队列已满，则会引发异常。

queue.offer(key); 将指定的元素插入队列。如果队列已满，则返回false。

queue.peek(); 返回堆顶元素，不出队。

queue.poll(); 返回堆顶元素，堆顶元素出队列。

queue.size(); 返回堆中元素个数。

queue.isEmpty(); 判断堆是否为空，为空返回true，不空返回false。

queue.remove(key); 从堆中删除指定的元素key。

queue.contains(key); 在堆中寻找指定的元素key。如果找到该元素返回true，否则返回false。

queue.toArray(); 将优先级队列转换为数组。并返回它。

1. 遍历

要遍历优先级队列中的元素，可以使用iterator()方法。queue.iterator();

**Num2710**

1. Java中去掉字符串尾部的0，str.replaceAll(“0+$”, ””);
2. Java中去掉字符串开头的0，str.replaceAll(“^0\*”, “”);

**Num724**

1、Java中计算int数组中所有元素的和：int[] nums = {1,2,3}; int sum = Arrays.stream(nums).sum();

**Num455**

1、int[] s = {1,3,2}; Arrays.sort(s);可以对s数组中的元素进行排序；