7-14:

队列Queue的用法:

LinkedList实现了Queue接口,所以可以将其视为队列

offer() => 队尾添加元素

poll() => 队头弹出元素

peek() => 返回第一个元素

反转一个数的牛皮思想

while (x != 0) {  
 n = n \* 10 + x % 10;  
 x = x / 10;  
}

7-15:

字符串的用法: subString()

二分查找的思想

while (low < high) {  
 int mid = (high - low + 1) / 2 + low;  
 if (prefix(strs, mid)) {  
 low = mid;  
 } else {  
 high = mid - 1;  
 }  
}

7-16:

map的containsKey() 法,判断map中是否有相应的key

栈 stack push() pop()

7-17:

数组转LinkedList: LinkedList linklist=new LinkedList(Arrays.asList(array));

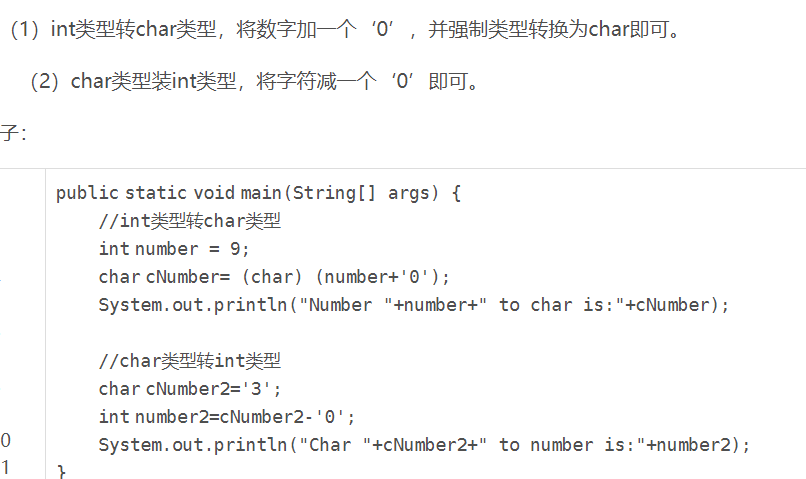
LinkedList转数组:

Integer[] arrray = (Integer[]) list.toArray(new Integer[list.size()]);

7-28

char字符有对应的ASCII码

两个char字符在做运算时会自动转换为对应的ACII码进行运算



(1)是利用了强转机制, char字符与数值加减,其char字符会转换为数值再计算

(2)是利用了ASCII码