**数据结构实验报告**

姓名： 赵楠 学号：U201617007 班级：软工1601班

**实验二**

1. 问题描述

1.、Radix Sort。实现桶式排序和基于桶式排序的基数排序。

2、Stack。用C语言设计堆栈，并实现中缀表达式到后缀表达式的转换。

二、问题分析与算法设计

1、问题1

桶式排序：将所有元素丢入一个大小为（max - min + 1 )的桶中，min对应桶中的下标为0，max对应桶中的下标为 (max - min)。

基数排序：

先按照优先级别最低的位对序列进行排序，下一步按照稍高优先级的位对序列进行排序，直到最高位时候整个序列就是有序序列。

2、问题2

通过分别对每个操作符设置一个优先级，让操作符根据优先级来判断来输出，具体请看实验方案。

1. 实验方案/步骤
2. 问题1

桶式排序方案如下：

（1）对于这个可枚举范围构建一个buckets数组，用于记录“落入”每个桶中元素的个数；

（2）将得到的buckets 数组中所有非零元素按照重复个数全部传入data（要进行排序的数组）中。

基数排序方案如下：

1. 问题2

优先级：) > \* / > + - > (

方案如下：

1）如果遇到操作数，我们就直接将其输出。

2）如果遇到操作符，首先检查操作符的优先级，若栈顶元素的优先级高于或等于操作符的优先级，那么就先将栈中元素弹出直到遇到发现更低优先级的元素(或者栈为空)为止。弹出完这些元素后，再将遇到的操作符压入到栈中。

3）遇到左操作符即’(‘，则直接压入栈中。

4）如果遇到一个右括号，将栈中的操作符弹出并输出直到遇到左括号为止。注意，左括号只弹出并不输出。

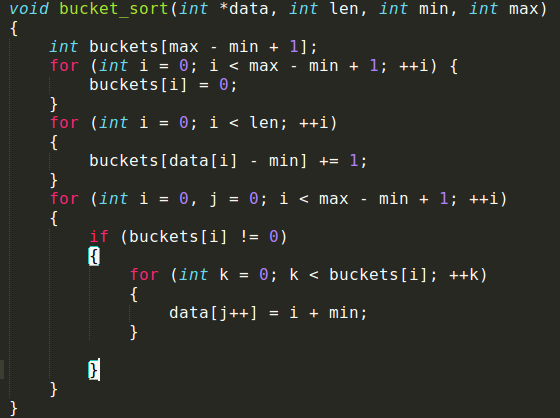
5）如果我们读到了输入的末尾，则将栈中所有元素依次弹出。

四、 算法实现

1、问题1

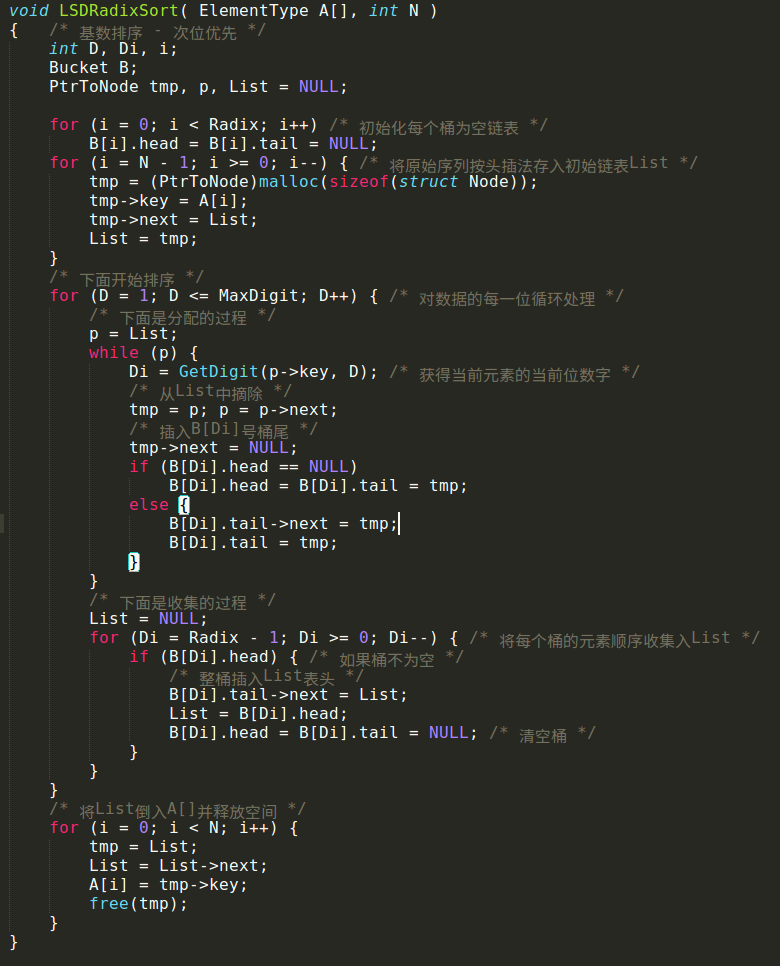
桶式排序：

关键函数：



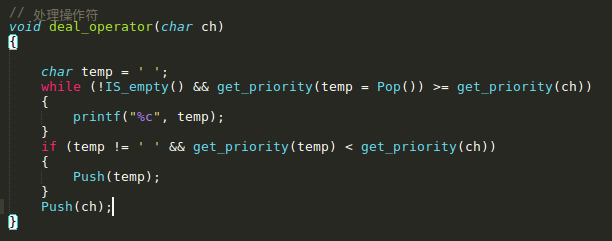
基数排序：

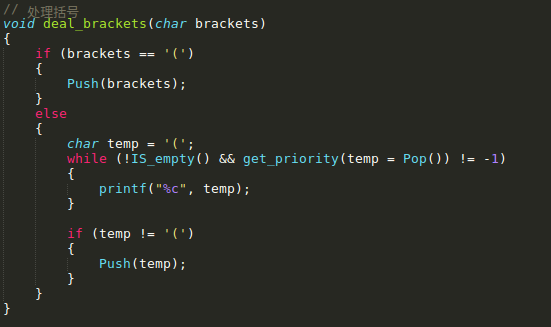
关键函数：



2、问题2

此处贴上关键函数代码（其他代码见文件stack.c）：





1. 测试结果与分析
2. 问题1

基数排序分析：

基数排序是基于分配策略的排序，不是一种比较排序，不受到 O(n log n) 下限的影响，是一种稳定的排序算法，并且它可以应用于多关键字排序。

1. 问题2

实例：

输入：1+2\*3+(4\*5+6)\*7

处理过程如下：

1. 读到 1，输出
2. 读到 + ，压入栈中
3. 读到2， 输出
4. 读到 \* ，检查栈中没有比 \* 高或者等于的操作符，于是压入栈中
5. 读到3， 输出
6. 读到 +， 由于栈中存在 高于或者等于 + 的优先级的操作符，所以首先弹出 \* + (同是输出），之后将读到的 + 压入栈中
7. 读到 ( ，压入栈中
8. 读到 4，输出
9. 读到 \*，检查没有比 \* 优先级高或者等于的操作符，所以压入栈中
10. 读到5 直接输出
11. 读到 + ，栈中 \* 优先级高于 +，输出 \*，将 + 压入栈中
12. 读到 6 输出
13. 读到 ），将栈中的元素输出知道遇到 （，即输出 +
14. 读到 \* ,压入栈中
15. 读到 7 ，直接输出
16. 最后将栈中的操作符全部输出，即输出 +

最后所有的输出为：1 2 3 \* + 4 5 \* 6 + 7 \* +