

## 目录

一、栈的应用.....	2
二、HuffmanTree.....	4
三、二叉树.....	5
四、二叉排序树 .....	7

ps: 主要包括题目和思路

# 一、栈的应用

```
#include <stack>
stack<int> S;
S.push(i);
S.pop();
int x = S.top();
```

## 例题：

题目描述：

在某个字符串（长度不超过 100）中有左括号、右括号和大小写字母；规定（与常见的算数式子一样）任何一个左括号都从内到外与在它右边且距离最近的右括号匹配。

写一个程序，找到无法匹配的左括号和右括号，输出原来字符串，并在下一行标出不能匹配的括号。不能匹配的左括号用"\$"标注,不能匹配的右括号用"?"标注.

输入：

输入包括多组数据，每组数据一行，包含一个字符串，只包含左右括号和大小写字母，字符串长度不超过 100。

输出：

对每组输出数据，输出两行，第一行包含原始输入字符，第二行由"\$","?"和空格组成，"\$"和"?"表示与之对应的左括号和右括号不能匹配。

样例输入：

```
)(rttyy())sss)(
```

样例输出：

```
)(rttyy())sss)(
```

```
?           ?$
```

思路：

多组输入->遇左括号索引入栈->遇右括号弹栈->输出

### 练习题：

时间限制：1s 内存限制：32M

（2006 年浙江大学机试）

题目描述：

读入一个只包含 +, -, \*, / 的非负整数计算表达式，计算该表达式的值。

输入：

测试输入包含若干测试用例，每个测试用例占一行，每行不超过 200 个字符，整数和运算符之间用一个空格分隔。没有非法表达式。当一行中只有 0 时输入结束，相应的结果不要输出。

输出：

对每个测试用例输出 1 行，即该表达式的值，精确到小数点后 2 位。

样例输入：

1 + 2

4 + 2 \* 5 - 7 / 11

0

样例输出：

3.00

13.36

思路：

传统方法：两个栈，一个保存运算符，一个保存数字

本题：由于不考虑括号，可取巧，仅使用一个栈（书上代码太多了。）

思路：

一个栈：

①若遇到加号把数字入栈；

②若遇到减号则将数字转换为负数后入栈；

③若遇到乘除则就近直接计算后，再把数值入栈；

④最后求和栈内的数字。按格式输出

ps: 见 github 源码：2\_1calculator.cpp

## 二、HuffmanTree

```
#include <queue>
using namespace std;
```

### 例题：

时间限制：1 秒 空间限制：65536K

哈夫曼树，第一行输入一个数  $n$ ，表示叶结点的个数。

需要用这些叶结点生成哈夫曼树，根据哈夫曼树的概念，这些结点有权值，即 **weight**，题目需要输出所有结点的值与权值的乘积之和。

输入描述：

输入有多组数据。

每组第一行输入一个数  $n$ ，接着输入  $n$  个叶节点（叶节点权值不超过 100， $2 \leq n \leq 1000$ ）。

输出描述：

输出权值。

示例 1

输入

5

1 2 2 5 9

输出

37

## 三、二叉树

### 例题：

#### 题目描述：

二叉树的前序、中序、后序遍历的定义：

前序遍历：对任一子树，先访问根，然后遍历其左子树，最后遍历其右子树；

中序遍历：对任一子树，先遍历其左子树，然后访问根，最后遍历其右子树；

后序遍历：对任一子树，先遍历其左子树，然后遍历其右子树，最后访问根。

给定一棵二叉树的前序遍历和中序遍历，求其后序遍历（提示：给定前序遍历与中序遍历能够唯一确定后序遍历）。

#### 输入：

两个字符串，其长度  $n$  均小于等于 26。

第一行为前序遍历，第二行为中序遍历。

二叉树中的结点名称以大写字母表示：A，B，C....最多 26 个结点。

#### 输出：

输入样例可能有多组，对于每组测试样例，

输出一行，为后序遍历的字符串。

#### 样例输入：

ABC

BAC

FDXEAG

XDEFAG

#### 样例输出：

BCA

XEDGAF

思路：还原树->后序遍历树

重点内容，遍历

前序：根，左，右

中序：左，根，右

后序：左，右，根

	根	左	右
前序	F   D	X E	A G
	左	根	右
中序	X D	E   F	A G
		左	右
后序			F
	递归		递归

对左：

	根	左	右
前序	D   X	E	
	左	根	右
中序	X   D	E	
	左	右	根
后序	X   E	D	

对右：

	根	左	右
前序	A   null	G	
	左	根	右
中序	null   A	G	
	左	右	根
后序	null   G	A	

因此可得：XEDGAF

## 四、二叉排序树

二叉排序树的定义：

二叉排序树的特点：

**例题：**

时间限制：1 秒 空间限制：32768K

题目描述

输入一系列整数，建立二叉排序树，并进行前序，中序，后序遍历。

输入描述：

输入第一行包括一个整数  $n(1 \leq n \leq 100)$ 。接下来的一行包括  $n$  个整数。

输出描述：

可能有多组测试数据，对于每组数据，将题目所给数据建立一个二叉排序树，并对二叉排序树进行前序、中序和后序遍历。

每种遍历结果输出一行。每行最后一个数据之后有一个空格。

输入中可能有重复元素，但是输出的二叉树遍历序列中重复元素不用输出。

示例 1

输入

5

1 6 5 9 8

输出

1 6 5 9 8

1 5 6 8 9

5 8 9 6 1

思路：排序树->遍历树

重点内容，遍历

前序：根，左，右

中序：左，根，右

后序：左，右，根

多组数据输入->生成排序树->遍历树（按格式输出）

### 练习题:

二叉搜索树

时间限制: 1 秒 空间限制: 65536K

判断两序列是否为同一二叉搜索树序列

输入描述:

开始一个数  $n$ , ( $1 \leq n \leq 20$ ) 表示有  $n$  个需要判断,  $n=0$  的时候输入结束。

接下去一行是一个序列, 序列长度小于 10, 包含(0~9)的数字, 没有重复数字, 根据这个序列可以构造出一颗二叉搜索树。

接下去的  $n$  行有  $n$  个序列, 每个序列格式跟第一个序列一样, 请判断这两个序列是否能组成同一颗二叉搜索树。

输出描述:

如果序列相同则输出 YES, 否则输出 NO

示例 1

输入

2

567432

543267

576342

0

输出

YES

NO

思路:

判断两个树的前序和中序(包含中序遍历的两种遍历即可)是否相等, 相等则 YES, 反之 NO

多组数据输入->插入生成排序树->对比前序和中序是否一致->输出结果

(代码和例题大致相同, 只需添加对比字符串 `strcmp` 即可)