# 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

# 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<=n<=1000）。

输出描述:

输出权值。

示例1

输入

5

1 2 2 5 9

输出

37