

高级程序设计与实验随堂测试十

1. (10分) 写一个 C++ 函数，输入是一个包含 n 个正整数的无序数组 `nums`（例如 `{6, 2, 3, 7}`），输出可以表达为这个数组中的某子集之和的所有整数，**从小到大**存放在一个整数向量中。在上面这个例子中，输出应为 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18.

已知 n 不超过 40，数组中最大的整数不超过 500，你的函数需要在一秒内运行完，否则最多得一半分。

函数格式: `vector<int> subsetsum(const vector<int>& nums);`

2. (7分)

写一个 C++ 函数，输入是一个字符串，判断其中的圆括号是否正确匹配。

函数格式：`bool parentheses(const string & s);`

若圆括号正确匹配，返回 `true`，否则返回 `false`

注意：只需判断圆括号 '(' 和 ')' 是否正确匹配即可。方括号和花括号无需判断

你的函数运行时间应为 $O(n)$ ，其中 n 是输入字符串的长度。否则最多只得一半分。

输入样例：

“3+5*(5/2)”

输出样例：

True

输入样例：

“3+(5*5)/2)”

输出样例：

False

输入样例：

“3+(5*5/(1+3)”

输出样例：

False

3. (13分) 给定 n 个正整数，它们代表一些木棍的长度（编号 $1, 2, \dots, n$ ），判断是否能用它们拼成一个正三角形。如果可以，打印它的三条边分别要用哪些编号的木棍；如果不可以，打印 NO。

写一个 C++ 函数完成这件事，它的输入是长度为 n 的正整数向量，输出直接打印到标准流(cout)

函数格式: `void triangle(const vector<int> & A);`

输入样例:

{3, 2, 7, 5, 1, 2, 1, 6}

输出样例:

1 4 5

2 3

6 7 8

样例解释:

三角形的三条边对应的木棍长度分别是{3, 5, 1}, {2, 7}, {2, 1, 6}