**课程报告**

**程序设计方法及实践(专创)**

**题　　目： 程序设计竞赛题目选做**

**作　　者： 　　梁浩铂**

**所在学院： 信息科学与工程学院**

**专业班级： 计算机专业21-1班**

**指导教师： 张 金 明**

**职　　称**：　　　 **讲 师**

**2022年12月18 日**

目 录

目录

[程序设计方法及实践(专创) I](#_Toc32727)

[第一章 程序设计入门 1](#_Toc719)

[1.1年份(year) 1](#_Toc16378)

[第二章 循环结构程序设计 2](#_Toc24305)

[2.1水仙花数(daffodil) 2](#_Toc17555)

[第三章 数组和字符串 3](#_Toc14923)

[3.1分子量(Molar Mass,ACM/ICPC Seoul2007,UVa1586) 3](#_Toc23225)

[第四章 函数和递归 5](#_Toc21600)

[4.1骰子涂色(Cube painting,UVa253) 5](#_Toc30593)

[4.2洪水!(Flooded!ACM/ICPCWorld Finals 1999,UVa815) 7](#_Toc15119)

[第五章 C++和STL入门 9](#_Toc4614)

[5.1交换学生(Foreign Exchange,UVa10763) 9](#_Toc28395)

[5.2对称轴(Symmetry,ACM/ICPC Seoul2004,UVa1595) 11](#_Toc7885)

[第六章 数据结构基础 12](#_Toc14243)

[6.1平衡的括号(Parentheses BalanceUVa 673) 13](#_Toc16159)

[6.2二叉树重建 (Tree Recovery,ULM1997UVa 536) 14](#_Toc4199)

[新疆大学课程论文（设计）、学年论文评分表 16](#_Toc3145)

## 程序设计入门

作业要求：在第一章的习题中任选一道

### 1.1年份(year)

#### 1.1.1 题目描述

输入年份，判断是否为闰年。如果是，则输出yes，否则输出no

提示:简单地判断除以4 的余数是不够的。

#### 1.1.2程序代码

#include <iostream>

using namespace std;

int main(){

int year;

cout<<"请输入要判断的年份："<<endl;

cin>>year;

if((year%4==0&&year%100!=0)||(year%400==0)){//能被4整除，但不能被100整除或者能被400整除的年份叫做闰年

cout<<"yes"<<endl;

}else{

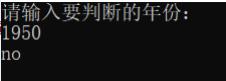
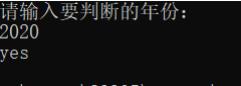
cout<<"no"<<endl;

}

return 0;

#### }

#### 1.1.3运行结果

#### 1.1.4分析总结

1. 使用cout、cin进行输出、输入
2. if(){... ...}else{... ...}判断语句
3. &&：逻辑与 ||：逻辑或
4. %：取模运算
5. 时间复杂度：O(1)

## 第二章 循环结构程序设计

作业要求：在第二章的习题中任选一道，例如：韩信点兵

### 2.1水仙花数(daffodil)

#### 2.1.1 题目描述

输出100~999中的所有水仙花数。若三位数ABC满足ABC=A3+B3+C3 ，则称其为水仙花数，例如153=13+53+33 ，所以153就是水仙花数。

#### 2.1.2程序代码

#include <iostream>

#include <cmath>

using namespace std;

int main() {

int a, b, c;

for (int i = 100; i < 1000; i++) {

c = i % 10;//个位数

b = (i / 10) % 10;//十位数

a = (i / 100) % 10;//百位数

int tmp = (int)pow(a, 3) + pow(b, 3) + pow(c, 3);//a^3+b^3+c^3

if (tmp == i) {

cout << i << endl;

}

else {

continue;

}

}

return 0;

#### }

#### 2.1.3运行结果

#### 

#### 2.1.4分析总结

1. for(..;...;...;)for循环语句
2. pow(...,...)函数，使用前需要引用头文件
3. if(){... ...}else{... ...}判断语句
4. 拆分数字
5. 时间复杂度O(n)

## 第三章 数组和字符串

作业要求：在第三章的习题中任选一道

### 3.1分子量(Molar Mass,ACM/ICPC Seoul2007,UVa1586)

#### 3.1.1 题目描述

给出一个物质的分子式(不带括号)，求分子量。本题中的分子式只包含4中原子，分别为C,H,O,N,原子 量为12.01,1.008,16.00 ， 14.01 (单位： g/mol)。例如， C6H5OH的分子量为94.108g/mol。

#### 3.1.2程序代码

#include <iostream>

#include <string>

using namespace std;

int main()

{

double arr[] = {12.01, 1.008, 16.00, 14.01};

string s;

cout << "请输入化学式：" << endl;

cin >> s;

int len = s.size();

double total = 0.0;

int i = 0;

while(s[i]!='\0'){

int sum = 0;

char c = s[i];

i++;

while (s[i] >= '0' && s[i] <= '9') {//计算脚标

sum = sum \* 10 + s[i] - '0';

i++;

}

if (sum != 0) {//没有脚标

if (c == 'C') {

total += sum \* arr[0];

}

else if (c == 'H') {

total += sum \* arr[1];

}

else if (c == 'O') {

total += sum \* arr[2];

}

else if (c == 'N') {

total += sum \* arr[3];

}

}

else{//存在脚标

if (c == 'C') {

total += arr[0];

}

else if (c == 'H') {

total += arr[1];

}

else if (c == 'O') {

total += arr[2];

}

else if (c == 'N') {

total += arr[3];

}

}

}

cout << "化学式对应的分子量为：" << endl;

cout << total << endl;

return 0;

#### }

#### 3.1.3运行结果

#### 

#### 3.1.4分析总结

1. 静态数组
2. while(...){... ...}循环语句
3. if(){... ...}else{... ...}判断语句
4. string容器
5. 时间复杂度：O(n)

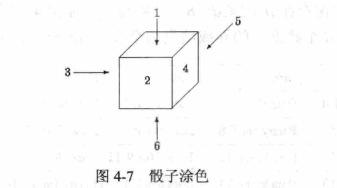
## 第四章 函数和递归

作业要求：在第四章的习题中任选两道

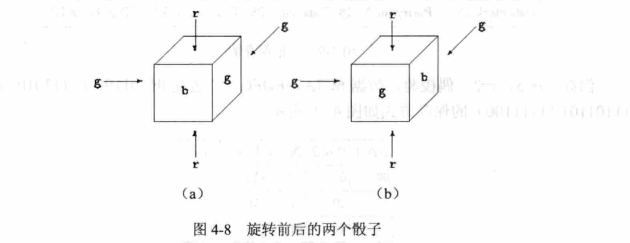
### 4.1骰子涂色(Cube painting,UVa253)

#### 4.1.1 题目描述

#### 输入两个散子，判断二者是否等价。每个骰子用6个字母表示，如图4-7所示。



例如rbgggr和rggbgr 分别表示如图48 所示的两个子。二者是等价的，因为图4-8 (a)所示的骰子沿着竖直轴旋转90°之后就可以得到图4-8()所示的骰子。



#### 4.1.2程序代码

#include<iostream>

#include <string>

using namespace std;

int main()

{

string a, b;

cout << "请输入骰子a：" << endl;

cin >> a;

cout << "请输入骰子b：" << endl;

cin >> b;

int cnt = 0;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 6; j++)

{

if (a[i] == b[j] && a[5 - i] == b[5 - j])//判断

{//骰子1、6相对，2、5相对，3、4相对

cnt++;

a[i] = a[5 - i] = '0';//避免重复判断

b[j] = b[5 - j] = '1';

break;

}

}

}

if (cnt == 3) {

cout << "TRUE" << endl;

}

else{

cout << "FALSE" << endl;

}

return 0;

#### }

#### 4.1.3运行结果

#### 

#### 4.1.4分析总结

1. for(..;...;...;)for循环语句
2. if(){... ...}else{... ...}判断语句
3. &&：逻辑与
4. a++:自增运算符（先自增，后赋值）
5. 时间复杂度为：O(n^2)

### 4.2洪水!(Flooded!ACM/ICPCWorld Finals 1999,UVa815)

#### 4.2.1 题目描述

有一个n\*m(1<=m,n<30)的网格，每个格子是变长10米的正方形，网格四周是无限大的墙壁。输入每个 格子的海拔高度，以及网格内雨水的总体积，输出水位的海拔高度以及有多少百分比的区域有水(即高 度严格小于水平面)。

#### 4.2.2程序代码

#include <iostream>

#include <vector>

#include <iomanip>

#include <algorithm>

using namespace std;

int main() {

int m, n;

int sum[1000] = { 0 };

double water, rate, high;

double avg[1000] = { 0.0 };

vector<int> height;

height.push\_back(0);

cout << "请输入m，n：" << endl;

cin >> m >> n;

cout << "请输入水量：" << endl;

cin >> water;

cout << "请输入每个格子的海拔高度：" << endl;

for (int i = 0; i < m \* n; i++) {

int h;

cin >> h;

height.push\_back(h);

}

sort(height.begin(), height.end());//把每个格子的海拔从小到大排序

sum[0] = 0;

int i = 1;

for (i = 1; i <= m \* n; i++) {

sum[i] = sum[i - 1] + height[i];

avg[i] = (double)sum[i] / i;//前n个格子的平均海拔高度

if (((double)height[i+1] - avg[i]) \* n \* 100 > water) {//计算前n个格子能否存储足够的水量

break;

}

}

rate = ((double)(i - 1)) / ((double)(m \* n));

high = (double)(water / 100 / (i - 1)) + avg[i - 1];//水的海拔高度

cout << "水位的海拔高度为：" << high << endl;

cout << "所占区域的百分比为：" << setiosflags(ios::fixed) << setprecision(2) << rate \* 100 << endl;

return 0;

}

#### 4.2.3运行结果

#### 

#### 4.2.4分析总结

1. vector容器
2. sort()排序
3. (int)a：强制转换，将a强制转换为int类型
4. Setprecision()：输出指定小数点后位数的函数
5. 时间复杂度为：O(n^2)

## 第五章 C++和STL入门

作业要求：在第五章的习题中任选两道

### 5.1交换学生(Foreign Exchange,UVa10763)

#### 5.1.1 题目描述

有n(1≤n≤500000)个学生想交换到其他学校学习。为了简单起见，规定每个想从A学校换到 B学校的学生必须找一个想从 B 换到A的“搭档”。如果每个人都能找到搭档(一个人不能当多个人的搭档)，学校就会同意他们交换。每个学生用两个整数 A、B 表示，你的任务是判断交换是否可以进行。

#### 5.1.2程序代码

#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

int main()

{

int n;

cout << "请输入学生个数：" << endl;

cin >> n;

vector<int> sa, sb;

for (int i = 0; i < n; i++)

{

int a, b;

cin >> a >> b;

sa.push\_back(a);

sb.push\_back(b);

}

sort(sa.begin(), sa.end());//排序

sort(sb.begin(), sb.end());//排序

int flag = 1;

for (int i = 0; i < n; i++)

{

if (sa[i] != sb[i])

{

flag = 0; break;

}

}

if (flag) {

cout << "是否可以交换:Yes" << endl;

}

else {

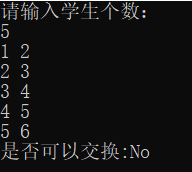
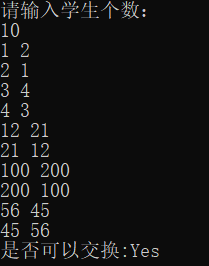
cout << "是否可以交换:No" << endl;

}

return 0;

}

#### 5.1.3运行结果

#### 5.1.4分析总结

1. vector容器
2. sort()排序
3. for(..;...;...;)for循环语句
4. if(){... ...}else{... ...}判断语句
5. 时间复杂度：O(n^2)

### 5.2对称轴(Symmetry,ACM/ICPC Seoul2004,UVa1595)

#### 5.2.1 题目描述

给出平面上N ( N<=1000)个点，问是否可以找到一条竖线，使得所有点左右对称。

#### 

#### 5.2.2程序代码

#include <iostream>

#include <set>

using namespace std;

int main() {

int n;

int sum = 0;

set<pair<int, int>> point;

bool flag = true;

cout << "请输入坐标点个数：" << endl;

cin >> n;

for (int i = 0; i < n; i++) {

int x, y;

cout << "请输入第" << i + 1 << "个点的x，y坐标：" << endl;

cin >> x >> y;

sum += x;

point.insert(pair<int, int>(n \* x, y));

}

for (set<pair<int, int>>::iterator i = point.begin(); i != point.end(); i++) {//迭代器遍历

pair<int, int> p = \*i;

if (point.find(pair<int,int> (2 \* sum - p.first,p.second)) == point.end()) {//判断对称

flag = false;

break;

}

}

if (flag) {

cout << "YES" << endl;

}

else {

cout << "NO" << endl;

}

return 0;

#### }

#### 5.2.3运行结果

#### 

#### 5.2.4分析总结

1. pair对组
2. set容器
3. iterator迭代器
4. if(){... ...}else{... ...}判断语句
5. 时间复杂度：O(n)

## 第六章 数据结构基础

作业要求：在第六章的习题中任选两道

### 6.1平衡的括号(Parentheses BalanceUVa 673)

#### 6.1.1 题目描述

#### 输入一个包含“0”和“”的括号序列，判断是否合法。具体规则如下:

1. 空串合法。
2. 如果A和B都合法，则AB 合法
3. 如果 A合法则(A)和[A]都合法。

#### 6.1.2程序代码

#include<iostream>

#include<stack>

#include <string>

using namespace std;

int main()

{

stack<char> str;

string s;

int flag = 1;

cout << "请输入括号：" << endl;

cin >> s;

for(char a : s) // 增强for

{

if (a == ')' && flag)

{

if (str.empty()) flag = 0;//判断‘()’

else if (str.top() == '(') str.pop();

else flag = 0;

}

else if (a == ']' && flag)

{

if (str.empty()) flag = 0;//判断‘[]’

else if (str.top() == '[') str.pop();

else flag = 0;

}

else str.push(a);

}

if (str.empty() && flag) {

cout << "Yes" << endl;

}

else {

cout << "No" << endl;

}

return 0;

#### }

#### 6.1.3运行结果

#### 

#### 6.1.4分析总结

1. string容器
2. stack容器
3. for(i : vector<int>)增强for，遍历容器
4. if(){... ...}else{... ...}判断语句
5. 时间复杂度：O(n)

### 6.2二叉树重建 (Tree Recovery,ULM1997UVa 536)

#### 6.2.1 题目描述

输入一棵二叉树的先序遍历和中序遍历序列，输出后序遍历序列

#### 

#### 6.2.2程序代码

#include <iostream>

#include <string>

using namespace std;

string preorder, inorder;

void get\_postorder(int s1,int s2,int length) {//递归函数

if (length == 0) {

return;

}

int len = inorder.find(preorder[s1]) - s2;

get\_postorder(s1 + 1, s2, len);//左子树

get\_postorder(s1 + len + 1, s2 + len + 1, length - len - 1);//右子树

cout << preorder[s1];

}

int main(){

int len;

cout << "二叉树的先序遍历结果为：" << endl;

cin >> preorder;

cout << "二叉树的中序遍历结果为：" << endl;

cin >> inorder;

len = inorder.size();

cout << "二叉树的后序遍历结果为：" << endl;

get\_postorder(0, 0, len);

return 0;

#### }

#### 6.2.3运行结果

#### image-20221212202622763 image-20221212202655434

#### 6.2.4分析总结

1. string容器
2. 函数递归
3. 全局变量
4. 空返回
5. 时间复杂度：O(n)

新疆大学课程论文（设计）、学年论文评分表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题 目 | 程序设计竞赛题目选做 | | | | |
| 作 者 | 梁浩铂 | 专业年级 | 计算机专业21-1班 | 指导教师 | 张金明 |
| 指导教师评语及  评分建议 | **指导教师评语：**  该课程论文写作格式 优美，良好，较差，代码格式凌乱，整齐，选做题目数量较少(<8)，足够(=8)，较多(>8)，题目类型单一、多样，题目难度一般，有挑战性，题目分析不全面，全面。  **评分建议：**  写作格式： 8 分； 代码质量： 8 分；  题目数量： 8 分； 题目类型： 8 分；  题目难度： 8 分； 题目解答： 8 分；  现场运行： 20 分； 提问答辩： 4\*8 分；  总计： 100 分；  指导教师：张金明  2022 年 12 月 18 日 | | | | |
| 院  （部）  或  教  研  室  意  见 | 学院或教研室主任：  年 月 日 | | | | |