**C++程序编程复习题**

**遍历**

**ch10\_11.cpp**

**#include <iostream>**

**#include <iomanip>**

**using namespace std;**

**struct Student**

**{**

**long number;**

**float score;**

**Student\* next;**

**};**

**Student\* head; //链首指针**

**Student\* getNode()**

**{**

**int num;**

**float sc;**

**cin>>num>>sc;**

**if(num==0)**

**return NULL;//节点无效，结束创建过程**

**Student\*p=new Student;**

**p->number=num;**

**p->score=sc;**

**return p;**

**}**

**void Create(){**

**if((head=getNode())==0)//新建第一个节点，挂入链首**

**return; //返回空列表**

**for(Student\*pE=head,\*pS;pS=getNode();pE=pS)//pE指向尾节点**

**pE->next=pS;}**

**void ShowList(){**

**cout<<"now the items of list are \n";**

**for(Student\*p=head;p;p=p->next)**

**cout<<p->number<<","<<p->score<<endl;}**

**int main(){**

**cout<<fixed<<setprecision(1);**

**Create();**

**ShowList();}**

**冒泡排序法**

**ch7\_9.cpp**

**#include <iostream>**

**using namespace std;**

**void bubble(int[],int);**

**int main()**

**{**

**int array[]={55,2,6,4,32,12,9,73,26,37};**

**int len=sizeof(array)/sizeof(int); //元素个数**

**for(int i=0; i<len; i++) //原始顺序输出**

**cout <<array[i] <<",";**

**cout <<endl<<endl;**

**bubble(array, len); //调用排序函数**

**}**

**void bubble(int a[],int size) //冒泡排序**

**{**

**int i,temp;**

**for(int pass=1; pass<size; pass++){ //共比较size-1轮**

**for(i=0; i<size-pass; i++) //比较一轮**

**if(a[i]>a[i+1]){**

**temp=a[i];**

**a[i]=a[i+1];**

**a[i+1]=temp;**

**}**

**for(i=0; i<size; i++) //比较一轮后就输出**

**cout <<a[i] <<",";**

**cout <<endl;**

**}**

**}**

**从键盘输入的整数中找出最大值，输入0时结束**

**#include <iostream>**

**using namespace std;**

**int main()**

**{int max=0,a;**

**do{cin>>a;**

**if(a>max) max=a;}**

**while(a!=0);**

**cout<<max;}**

**斐波那契数列**

**练习5.5**

**#include <iostream>**

**using namespace std;**

**int fib(int n);**

**int fib\_1(int n);**

**int main()**

**{**

**int n;**

**cout<<"想求数列中的第几项？\n";**

**cin>>n;**

**while(n<=0)**

**{**

**cout<<"请输入大于0的整数！"<<endl;**

**cin>>n;**

**}**

**cout<<"第"<<n<<"项斐波那契数列的值为："<<fib(n)<<endl;**

**cout<<"第"<<n<<"项斐波那契数列的值为："<<fib\_1(n)<<endl;**

**return 0;**

**}**

**int fib(int n)//非递归函数调用形式**

**{**

**int f1,f2,fn;**

**f1=f2=1;**

**for(int i=3;i<=n;i++)**

**{**

**fn=f1+f2;**

**f1=f2;**

**f2=fn;**

**}**

**return fn;**

**}**

**int fib\_1(int n)//递归函数调用形式**

**{**

**if(n==1||n==2)**

**return 1;**

**else**

**return fib\_1(n-1)+fib\_1(n-2);**

}

**水仙花数**

**练习4.3**

**例：求100－999之间的水仙花数**

**说明：水仙花数是指一个三位数的各位数字的立方和是这个数本身。**

**例如：（153＝1\*1\*1＋5\*5\*5＋3\*3\*3）**

**#include <iostream>**

**using namespace std;**

**int fun(int n)**

**{ int a,b,c;**

**a=n/100;**

**b=n/10%10;**

**c=n%10;**

**if(a\*a\*a+b\*b\*b+c\*c\*c==n) //或用pow（camth头文件）来表示**

**return 1；**

**else**

**return(0);**

**}**

**int main()**

**{**

**int i;**

**for(i=100;i<1000;i++)**

**if(fun(i)==1)**

**cout<<i<<" is ok!\n";**

**}**

**求n的阶乘**

**通过函数的递归调用计算阶乘**

**#include <iostream>**

**using namespace std;**

**long power(int n)**

**{**

**long f;**

**if(n==1||n==0)**

**f=1;**

**else if(n>1)**

**f=n\*power(n-1);**

**return (f);}**

**int main( ){**

**int n;**

**long y;**

**cout<<"input a inteager number:\n";**

**cin>>n;**

**y=power(n);**

**cout<<n<<"!="<<y<<"\n";}**

**判明素数**

**1：m是素数的条件是不能被2，3，…m-1整除**

**ch4\_6**

**#include <iostream>**

**int main()**

**{//输入**

**long m;**

**cout <<"please input a number:\n";**

**cin >>m;**

**//处理**

**int i;**

**for(i=2; i<m; i++) //找m的因数**

**if(m%i==0)**

**break;//输出**

**if(m==i) //判断m是否被小于m的数整除**

**cout <<m <<" is prime.\n";**

**else**

**cout <<m <<" isn't prime.\n";}**

**2：假定m不是素数，则可以表示为m=i\*j,i<=j,则i<=sqrt(m),j>=sqrt(m)**

**也就是说，如果m不是素数，一定能在sqrt（m）内找到一个整数i能整除m，即m%i=0**

**于是，循环可以在2~sqrt(m)中进行**

**ch4\_7**

**#include <iostream>**

**#include <cmath>**

**int main()**

**{//输入**

**long m;**

**cout <<"please input a number:\n";**

**cin >>m;//处理**

**double sqrtm=sqrt(m); //用到math.h**

**int i;**

**for(i=2; i<=sqrtm; i++)**

**if(m%i==0)**

**break;//输出**

**if(sqrtm<i)**

**cout <<m <<" is prime.\n";**

**else**

**cout <<m <<" isn't prime.\n";}**

**事实上，a~b的循环步长可以是2，因为偶数不是素数。单循环前要判明a是否为偶数**

**ch4\_8**

**#include <iostream>**

**#include <iomanip>**

**#include <cmath>**

**int main()**

**{//输入**

**long a,b,l=0;**

**cout <<"please input two numbers:\n";**

**cin >>a >>b;**

**cout <<"primes from " <<a <<" to " <<b <<" is:\n";**

**//处理**

**if(a%2==0) //若为偶数，则增1**

**a++;**

**for(long m=a; m<=b; m+=2)**

**{//步长为2**

**int sqrtm=sqrt(m);**

**int i;**

**for(i=2; i<=sqrtm; i++)**

**{//判明素数**

**if(m%i==0)**

**break;//输出**

**if(i>sqrtm)**

**{ //素数**

**if(l++%10==0)**

**cout <<endl;**

**cout <<setw(5) <<m;} }**

**}**

**调用函数形式确定素数**

**#include <iostream>**

**#include <iomanip>**

**#include <cmath>**

**bool isprime(long n);**

**int main()**

**{//input**

**long a,b,l=0;**

**cout<<"please input two numbers:\n";**

**cin >>a >>b;**

**cout <<"primes from " <<a <<" to " <<b <<" is \n";//process**

**if(a%2==0) a++;**

**for(long m=a; m<=b; m+=2)**

**if(isprime(m)){**

**//output**

**if(l++%10==0)**

**cout <<endl;**

**cout <<setw(5) <<m;**

**}**

**}**

**bool isprime(long n)**

**{**

**int sqrtm=sqrt(n);**

**for(int i=2; i<=sqrt(m); i++) //判明素数**

**if(n%i==0)**

**return false;**

**return true;**

**}**

**完数**

**练习4.4**

**完数的概念：指一个数恰好等于它的因子之和，如6=1+2+3**

**例：求1000内的完数**

**#include <iostream>**

**using namespace std;**

**int main(){**

**for(int i=1;i<=1000;i++)**

**{**

**int sum=0;//sum的初始化很重要**

**for(int j=1;j<=i/2;j++) //j遍历到一半即可**

**{**

**if(i%j==0)**

**sum+=j;//sum=sum+j; }**

**if(sum == i)**

**cout<<i<<"是完数"<<endl;}**

**return 0;**

**}**

**闰年**

**年份year为闰年的条件为：**

1. **能够被4整除，但不能被100整除的年份**

**②能够被400整除的年份设定标志变量leap，只要符合其中一个条件的就是闰年，令 leap =1；否则令 leap=0**

**如何判断year能否被4整除？**

**year%4==0 能整除 year%100！=0 不能整除**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int year, leap ;**

**cin>> year;**

**if(year%4 == 0 && year%100 != 0) leap=1;**

**else if(year%400 == 0) leap=1;**

**else leap=0;**

**if(leap == 1)**

**cout << year << “ is a leap year " <<endl;**

**else**

**cout << year << “ is not a leap year " <<endl;**

**return 0;**

**}**

**用递归来求两个整数的最大公约数（P90）**

**#include <iostream>**

**using namespace std;**

**long d1(int a,int b);**

**long d2(int a,int b);**

**int main()**

**{ int x,y;**

**cin>>x>>y;**

**cout<<"x,y的最大公约数为："<< d1(x,y)<<endl;**

**cout<<"y,x的最大公约数为："<< d2(x,y)<<endl; }**

**long d1(int a,int b){//递归版**

**if(a%b==0) return b;**

**return d1(b,a%b); }**

**long d2(int a,int b){//非递归版**

**int temp;**

**while(b!=0)**

**{temp=a%b;**

**a=b;**

**b=temp;**

**}**

**return a; }**

**百马百担问题**

**有100匹马，驮100担货，大马驮三担，中马驮两担，两匹小马驮一担**

**求大、中、小马各多少匹？**

**#include <iostream>**

**using namespace std;**

**int main( )**

**{**

**int hb,hm,hl;**

**int n=0;**

**for(hb=0;hb<=100;hb+=3)**

**for(hm=0;hm<=100-hb;hm+=2)**

**{hl=100-hb-hm;**

**if(hb/3+hm/2+2\*hl==100)**

**{n++;**

**cout<<"hb="<<hb/3**

**<<",hm="<<hm/2**

**<<",hl="<<2\*hl<<"\n";}**

**}**

**cout<<"n="<<n<<"\n";**

**}**

**百钱买百鸡问题**

**给定100元，要求买100只鸡，已知公鸡5元一只，母鸡3元一只，小鸡1元3只，问公鸡母鸡和小鸡应该各买几只？**

**#include <iostream>**

**using namespace std;**

**int main(){**

**int x,y,z;**

**for(x=1;x<=20;x++)**

**for(y=1;y<=33;y++)**

**{z=100-x-y;**

**if(5\*x+3\*y+z/3==100)**

**cout<<x<<”\t”<y<<”\t”<<z<<endl;}**

**}**

**抽球问题：**

**下面的功能是从三个红球，五个白球，六个黑球中任意取出八个球，且其中必需有白球，输出可能结果**

**#include <iostream>**

**using namespace std;**

**int main(){**

**int i,j,k;**

**cout<<" hong bai hei "<<endl;**

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

**for(j=1;j<=5;j++)**

**{**

**k=8-i-j;**

**if(k<6)**

**cout<<i<<""<<j<<""<<k<<endl;}**

**}**

**题目：将两个字符串连接为一个字符串，不许使用库函数strcat**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**void JOIN(char s1[],char s2[])**

**{**

**int i,j;**

**j=strlen(s1);**

**for (i=0; s2[i]!='\0';i++)**

**s1[i+j]=s2[i];**

**s1[i+j]='\0';**

**}**

**int main()**

**{**

**char str1[80],str2[40];**

**gets(str1);gets(str2);**

**puts(str1);puts(str2);**

**JOIN(str1,str2);**

**puts(str1);**

**}**

**阿克曼函数（Ackerman）**

**是非原始递归函数的例子。它需要两个自然数作为输入值，输出一个自然数。它的输出值增长速度非常高，仅是对于(4,3)的输出已大得不能准确计算。**

**#include <iostream>**

**using namespace std;**

**int ack(int m,int n)**

**{if(m==0)**

**return(n+1);**

**else if(n==0)**

**return(ack(m-1,1));**

**else**

**return(ack(m-1,ack(m,n-1)));**

**}//阿克曼函数的定义**

**int main()**

**{**

**int m,n;**

**cin>>m>>n;**

**cout<<ack(m,n)<<endl;**

**}**

**输出杨辉三角形**

**#include<iostream>**

**#include<iomanip>**

**using namespace std;**

**int\* YangHuiData(int n); //n是杨辉三角形的行数**

**void YangHuiShow(int \*pData, int n);//在屏幕上打印杨辉三角形**

**int main() {**

**int n;**

**cout << "请输入杨辉三角形的行数";**

**cin >> n;//计算杨辉三角并保存到二维数组**

**int \*pData = YangHuiData(n);//在屏幕上打印等腰三角形的杨辉三角**

**YangHuiShow(pData, n);//n是杨辉三角形的行数**

**}**

**int\* YangHuiData(int n) {**

**int \*a = new int[n\*n];//完成杨辉三角形的计算，没有显示的位置填0**

**int i, j;**

**for (i = 0; i<n\*n; i++) {a[i] = 0;}**

**a[0] = 1;a[n] = 1;a[n + 1] = 1;**

**for (i = 2; i<n; i++)**

**{a[i\*n] = 1;a[i\*n + i] = 1;**

**for (j = 1; j<i; j++)**

**{a[i\*n + j] = a[(i - 1)\*n + j - 1] + a[(i - 1)\*n + j];}**

**}**

**return a;**

**}**

**void YangHuiShow(int \*a, int n) {**

**//根据数据a指向的二维数组在屏幕上打印**

**//杨辉三角形，每行前面填充合适的空格**

**//以显示成等腰三角形，左右对称**

**int i, j;**

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

**{for (j = 0; j<n - i - 1; j++)**

**cout << setw(3) << " ";//前导空格，为单个数据的一半宽度**

**for (j = 0; j <= i; j++)**

**cout << setw(6) << a[i\*n + j];**

**cout << endl;}**

**}**

**杨辉三角的简单版行数：**

**#include <iostream>**

**#include <iomanip>**

**using namespace std;**

**int main()**

**{**

**int i,j,k,m,n;**

**int a[8][8];//输出的行数不能太多哟**

**for(i=0;i<8;i++)**

**{**

**a[i][0]=1;**

**a[i][i]=1;**

**}**

**for(i=2;i<8;i++)**

**{**

**for(j=1;j<=i-1;j++)**

**a[i][j]=a[i-1][j-1]+a[i-1][j];**

**}**

**for(i=1;i<8;i++)**

**{**

**for(k=1;k<8-i;k++)//控制输出的空格数**

**cout<<setw(5)<<" ";**

**for(m=i-1;m<i;m++)**

**{**

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

**cout<<setw(5)<<a[m][n]<<setw(5)<<" ";**

**}**

**cout<<endl;**

**}**

**return 0;**

**}**

**题目：输入一串字符串，然后倒序输出**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**int main()**

**{**

**char st[60],ch;**

**int i,j,len;**

**cin>>st;**

**len=strlen(st);**

**for(i=0,j=len-1;j>0;i++,j--)**

**{**

**ch=st[i];**

**st[i]=st[j];**

**st[j]=ch;}**

**cout<<st<<endl;**

**}**

**求最小公倍数和最大公约数**

**#include <iostream>**

**using namespace std;**

**//辗转相除法**

**int maxnumber(int a, int b) {**

**int t;**

**while (b != 0) {**

**t = a % b;**

**a = b;**

**b = t;**

**}**

**return a;**

**}**

**int minnumber(int a, int b) {**

**int t, m, n;**

**m = a; n = b;**

**while (b != 0) {**

**t = a % b;**

**a = b;**

**b = t;**

**}**

**return m \* n / a;**

**}**

**//辗转相减法**

**int maxnumber(int a, int b) {**

**int t;**

**while (a!= b) {**

**if (a > b)**

**a = a - b;**

**else**

**b = b - a;**

**}**

**return a;}**

**int minnumber(int a, int b) {**

**int t, m, n;**

**m = a; n = b;**

**while (a != b) {**

**if (a > b)**

**a = a - b;**

**else**

**b = b - a;}**

**return m \* n / a;**

**}**

**//穷举法**

**int maxnumber(int a,int b) {**

**int m, n,i;**

**m = a; n = b;**

**for (i = 1; i <= a; i++)**

**{**

**if (a%i == 0 && b%i == 0)**

**break;**

**}**

**for (i = a; i > 0; i--)**

**{**

**if (a%i == 0 && b%i == 0)**

**break;**

**}**

**return i;**

**}**

**int minnumber(int a, int b) {**

**int m, n,i;**

**m = a; n = b;**

**for (i = 1; i <= a; i++)**

**{**

**if (a%i == 0 && b%i == 0)**

**break;**

**}**

**for (i = a; i > 0; i--)**

**{**

**if (a%i == 0 && b%i == 0)**

**break;**

**}**

**return m \* n / i;**

**}**

**int main(){int a, b; cin >> a >> b;//定义两个函数的方法**

**cout << "最大公约数：" << maxnumber(a, b) << endl;**

**cout << "最小公倍数：" << minnumber(a, b) << endl;**

**return 0;}**

**回文**

**判断一个数是否为回文数，有多种方法，常见的有三种：**

**1.将输入数字转化为字符串。回文数关于中心对称，只需比较对称的数是否相等即可。**

**2.采用除10取余的方法，从最低位开始，依次取出该数的各位数字，然后用最低为充当最高位，按反序构成新的数，再与原数比较。**

**3.采用栈的方式。判断出栈的元素与栈内字符是否相等。**

**例：法2：**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**bool hui(int n);**

**int a;**

**while(cin>>a)**

**{**

**if(hui(a))**

**cout<<"true"<<endl;**

**else**

**cout<<"false"<<endl;**

**}**

**return 0;**

**}**

**bool hui(int n)**

**{**

**unsigned i=n;**

**unsigned m=0;**

**while(i>0)**

**{**

**m=m\*10+i%10;**

**i/=10;**

**}**

**return m==n;**

**}**

**求一个区间内的回文数：**

**#include <iostream>**

**using namespace std;**

**bool Palindrome(int x);**

**int main()**

**{**

**for (int i = 11; i <= 999; i++)**

**{**

**if (Palindrome(i) && Palindrome(i \* i) && Palindrome(i \* i \* i))**

**cout << i << endl; }**

**return 0;}**

**bool Palindrome(int x)**

**{**

**int n = x;**

**int a = 0;**

**while (n > 0)**

**{**

**a = a \* 10 + n % 10; //颠倒比较**

**n = n / 10;**

**}**

**return a == x;**

**}**

**题目：凡用过C语言标准库函数strcpy(char \*s1,char \*s2)的程序员都知道使用该函数时有一个安全隐患，即当指针s1所指向的空间不能容纳字符串s2的内容时，将发生内存错误。**

**类String的Strcpy成员函数能进行简单的动态内存管理，其内存管理策略为：**

**(1)若已有空间能容纳下新字符串，则直接进行字符串拷贝；**

**(2)若已有空间不够时，它将重新申请一块块内存空间(能容纳下新字符串)，把那个将新字符串内容拷贝到新申请的空间中，释放原字符串空间。**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**class String**

**{**

**private:**

**int size; // 缓冲区大小**

**char \*buf;// 缓冲区**

**public:**

**String(int bufsize);**

**void Strcpy(char \*s);**

**// 将字符串s复制到buf中**

**void Print() const;**

**~String()**

**{**

**if (buf != NULL)**

**delete [] buf;**

**}**

**};**

**String::String(int bufsize)**

**{**

**size = bufsize;**

**buf = new char[size];**

**\*buf = '\0';**

**}**

**void String::Strcpy(char \*s)**

**{**

**char \*p,\*q;**

**int len = strlen(s);**

**if (len+1 > size)**

**// 缓冲区空间不够，须安排更大空间**

**{**

**size = len+1;**

**p = q = new char[size];**

**while((\*q=\*s)!=0)// TODO: 添加代码将字符串s拷贝到字符指针q中**

**{q++；s++；}**

**delete [] buf;**

**buf = p;**

**}**

**else**

**{**

**for(p=buf;\*p=\*s;p++,s++);**

**// TODO: 添加代码将字符串s拷贝到buf中**

**}**

**}**

**void String::Print() const**

**{**

**cout << size << '\t' << buf << endl;**

**}**

**int main()**

**{**

**char s[100];**

**String str(32);**

**cin.getline(s, 99);**

**str.Strcpy(s);**

**str.Print();**

**return 0;**

**}**

**题目：对任何一个自然数n，如果它是偶数，那么把它砍掉一半；如果它是奇数，那么把(3n+1)砍掉一半。这样一直反复砍下去，最后一定在某一步得到n=1。卡拉兹在1950年的世界数学家大会上公布了这个猜想，传说当时耶鲁大学师生齐动员，拼命想证明这个貌似很傻很天真的命题，结果闹得学生们无心学业，一心只证(3n+1)，以至于有人说这是一个阴谋，卡拉兹是在蓄意延缓美国数学界教学与科研的进展……**

**我们今天的题目不是证明卡拉兹猜想，而是对给定的任一不超过1000的正整数n，简单地数一下，需要多少步（砍几下）才能得到n=1？**

**输入格式：每个测试输入包含1个测试用例，即给出自然数n的值。**

**输出格式：输出从n计算到1需要的步数。**

**输入样例：**

**3**

**输出样例：**

**5**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**int func(int n){**

**int i;**

**for(i=0;n!=1;i++)**

**{**

**if(n%2==0) n=n/2;**

**else if(n%2!=0)**

**n=(3\*n+1)/2;}**

**return i;**

**}**

**int main()**

**{**

**int n;**

**cin>>n;**

**cout<<func(n)<<endl;**

**return 0;**

**}**

**题目：编程求出将纸币（<=20元的整数，不考虑5角）兑换成1分、2分或5分硬币，共有多少种不同的兑换方法。**

**输入格式:**

**输入纸币金额。**

**输出格式:**

**输出兑换方法数，没有其它任何附加字符。**

**输人样例:**

**1**

**输出样例:**

**541**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**int func(int money)**

**{**

**int a,b,c,n=0;**

**for(a=0;a<=20\*money;a++)**

**for(b=0;b<=50\*money;b++)**

**for(c=0;c<=100\*money;c++)**

**if(5\*a+2\*b+c==money\*100)**

**n++;**

**return n;**

**}**

**int main()**

**{**

**int m;**

**cout<<"请输入钱数（单位：元）：";**

**cin>>m;**

**cout<<func(m)<<endl;**

**return 0;**

**}**

**题目：编写一个函数，返回数组中非负数据的平均值，假定数组中非负数据的个数不少于1个。要求以数组及数组元素个数作为函数参数。**

**#include<iostream>**

**using namespace std;**

**double ave(int array[], int size);**

**const int N = 10;**

**int main()**

**{**

**int a[N] = {50,-60,80,90,95,-100,85,88,-75,78};**

**double average =ave(a,N);**

**cout << average << endl;**

**return 0;**

**}**

**double ave(int array[], int size)**

**{**

**double sum=0;**

**int n=0;**

**for(int i=0;i<size;i++)**

**if(array[i]>=0)**

**{**

**sum+=array[i];**

**n++;}**

**return sum/n;**

**}**

**题目：两个乒乓球队进行比赛，各出三人。甲队为a,b,c三人，乙队为x,y,z三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a说他不和x比，c说他不和x,z比，请编程序找出三队赛手的名单。**

**#include <iostream>**

**using namespace std;**

**int main()**

**{char i,j,k;**

**/\*i是a的对手，j是b的对手，k是c的对手\*/**

**for(i='x';i<='z';i++)**

**for(j='x';j<='z';j++)**

**{if(i!=j)//这里放一个判断，比赛队员不能重复比赛，判断写在外面又可以减少循环次数**

**for(k='x';k<='z';k++)**

**{**

**if(i!=k && j!=k)//同上，排除相同**

**{**

**if(i!='x'&&k!='x'&&k!='z')//这里根据题意排除掉不符合的名单**

**cout<<"order is a--"<<i**

**<<"\tb--"<<j**

**<<"\tc--"<<k<<"\n";**

**}**

**}**

**}**

**}**

**题目：函数delnum的功能是删除一个字符串中的所有数字字符。**

**如输入："abc12d3f4",则输出："abcdf"**

**#include <iostream>**

**using namespace std;**

**void delnum(char \*s)**

**{**

**int i,j;**

**for(i=0,j=0; s[i]!='\0' ;i++)**

**if(s[i]<'0'||s[i]>'9')**

**{**

**s[j]=s[i];**

**j++;**

**}**

**s[j]='\0';**

**}**

**int main()**

**{**

**char item[100];**

**cout<<"\n input a string:\n";**

**gets(item);**

**delnum(item);**

**cout<<"\n"<<item;**

**}**

**题目：统计一个字符串中的字母、数字、空格和其它字符的个数。**

**#include <iostream>**

**using namespace std;**

**void fun(char s[],int b[])**

**{**

**int i;**

**for (i=0;s[i]!='\0';i++)**

**if('a'<=s[i]&&s[i]<='z'||'A'<=s[i]&&s[i]<='Z')**

**b[0]++;**

**else if (s[i]>='0'&&s[i]<='9')**

**b[1]++;**

**else if (s[i]==' ')**

**b[2]++;**

**else**

**b[3]=b[3]+1;**

**}**

**int main()**

**{**

**char s1[80];int a[4]={0};**

**int k;**

**gets(s1);**

**fun(s1,a);**

**puts(s1);**

**for(k=0;k<4;k++)**

**cout<<a[k]<<"\n";**

**}**

**题目：求1!+3!+5!+……+n!的和**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**long int f,s=0;**

**int i,j,n;**

**cin>>n;**

**for(i=1;i<=n;i=i+2)**

**{**

**f=1;**

**for(j=1;j<=i;j++)**

**f\*=j;**

**s=s+f;**

**}**

**cout<<"n="<<n**

**<<",s="<<s<<"\n";**

**}**

**变式题：求1!+2!+3!+……+n!的和**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int sum=0,temp=1;**

**//double sum=0,temp=1;**

**int i,n;**

**cin>>n;**

**for (i=1;i<=n;i++)**

**{**

**temp\*=i;**

**sum+=temp;**

**}**

**cout<<"n="<<n<<"sum="<<sum**

**<<endl;**

**return 0;**

**}**

**题目：计算某日是当年的第几天**

**提示：**

**（1）能被4整除但是不能被100整除的是闰年**

**（2）能被400整除的是闰年**

**#include <iostream>**

**using namespace std;**

**struct**

**{**

**int year;**

**int month;**

**int day;**

**}data;**

**/\* 定义一个结构并声明对象为data \*/**

**int main()**

**{**

**int days;**

**cout<<"请输入日期(年,月,日)：";**

**cin>>data.year>>data.month>>data.day;**

**switch(data.month)**

**{**

**case 1:days = data.day;**

**break;**

**case 2:days = data.day+31;**

**break;**

**case 3:days = data.day+59;**

**break;**

**case 4:days = data.day+90;**

**break;**

**case 5:days = data.day+120;**

**break;**

**case 6:days = data.day+151;**

**break;**

**case 7:days = data.day+181;**

**break;**

**case 8:days = data.day+212;**

**break;**

**case 9:days = data.day+243;**

**break;**

**case 10:days = data.day+273;**

**break;**

**case 11:days = data.day+304;**

**break;**

**case 12:days = data.day+334;**

**break;**

**}**

**if((data.year%4==0&&data.year%100!=0) ||(data.year%400==0))**

**if(data.month>=3)**

**days++;**

**cout<<data.month<<"月"**

**<<data.day<<"日是"**

**<<data.year<<"年的第"**

**<<days<<"天.\n";**

**}**

**题目：输入一个5个字符组成的字符串，将其倒序后输出.**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,j;**

**char st[6],ch;**

**cin>>st;**

**for(i=0,j=4;j>i; i++,j--)**

**{ch=st[i];**

**st[i]=st[j];**

**st[j]=ch;**

**}**

**cout<<st<<endl;**

**}**

**[题目] 定义一个类prime，用于求在一个指定的范围内素数的个数。**

**例如：在[3，50]范围内的素数有14个**

**[编程要求]**

**（1）私有数据成员。**

**int low：存放给定整数范围的下限**

**int up：存放给定整数范围的上限**

**int count：存放满足条件的整数的个数**

**（2）公有成员函数**

**prime(int n1,int n2)构造函数，用参数n1,n2初始化下限low和上限up,count的初值为0；**

**void process（）：在[low，up]范围中找出素数的个数，并把值赋给count;**

**void print（）：输出素数的个数.**

**（3）在主函数中定义prime类的对象p,给定的范围为[3，50]，**

**通过对象p调用成员函数实现计算及输出调试。**

**本题要求的输出结果为：[3，50]范围内有14个素数。**

**#include<iostream>**

**#include <cmath>**

**#include<fstream>**

**using namespace std;**

**class prime{**

**int low;**

**int up;**

**int count;**

**public:**

**prime(int n1,int n2)**

**{low=n1;up=n2;count=0;}**

**void process()**

**{**

**for(int i=low;i<=up;i++)**

**{**

**for(int j=2;j<i;j++)**

**if(i%j==0)break;**

**if(j==i) count++; }**

**void print()**

**{**

**cout<<'['<<low<<','<<up<<"]范围内有"<<count<<"个素数\n";**

**//此处将结果输出到文件"bc02.in"，请勿改动，否则影响判分**

**ofstream outf("bc02.in");**

**outf<<'['<<low<<','<<up<<"]范围内有"<<count<<"个素数\n";**

**outf.close();}**

**};**

**int main()**

**{**

**prime p(3,50);**

**p.process();**

**p.print();**

**return 0;**

**}**

**题目：输入20个整数到一维数组，统计正整数的个数及其和**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int a[20],i,sum,count;**

**sum=count=0；**

**for(i=0;i<20;i++)**

**cin>>a[i];**

**for(i=0;i<20;i++)**

**{**

**if( a[i]>0 )**

**{ \_\_\_\_\_\_count+=1; \_\_\_\_\_\_\_\_\_ //计数**

**\_\_\_\_\_\_sum+=a[i]; \_\_\_\_\_\_\_\_\_ //求和**

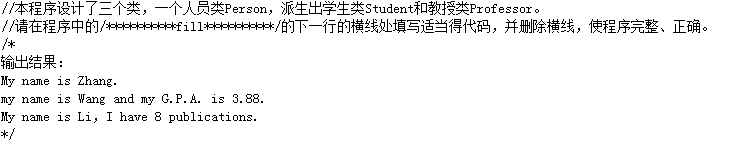
**}**

**}**

**cout<<count<<" "<<sum<<endl;**

**return 0;**

**}**

****

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**class Person{**

**public:**

**\_\_\_\_ Person()\_\_{name=NULL;}**

**Person(char\* s)**

**{**

**name = new char[strlen(s)+1]; strcpy(name, s);**

**}**

**~Person()**

**{**

**if(name!=NULL) delete [] name ;**

**}**

**\_\_\_\_ virtual void\_\_\_\_\_\_\_\_Disp() // 声明虚函数**

**{**

**cout << "My name is " << name << ".\n";**

**}**

**void setName(char\* s)**

**{**

**name = new char[strlen(s)+1]; strcpy(name, s);**

**}**

**protected:**

**char\* name;**

**};**

**class Student : public Person{**

**public:**

**Student(char\* s, double g) \_\_:Person(s), gpa(g)\_\_{ }**

**void Disp()**

**{**

**cout << "my name is " << name <<**

**" and my G.P.A. is " << gpa << ".\n";**

**}**

**private:**

**float gpa;**

**};**

**class Professor : public Person{**

**public:**

**void setPubls(int n){publs=n; }**

**void Disp()**

**{**

**cout<< "My name is "<<name<<"，I have " << publs << " publications.\n";**

**}**

**private:**

**int publs;**

**};**

**int main()**

**{ Person\* p\_\_\_\_\_\_\_\_\_;**

**Person x("Zhang");**

**p = &x; p->Disp();**

**Student y("Wang", 3.88);**

**p = &y; p->Disp();**

**Professor z;**

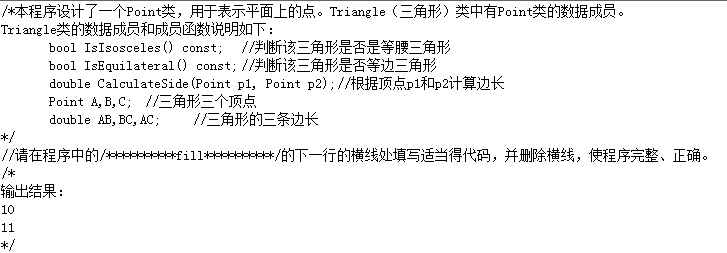
**z.setName("Li");**

**z.setPubls(8);**

**p = &z; p->Disp();**

**return 0;**

**}**



**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**#define MINNUM 0.00001**

**class Point**

**{**

**public:**

**Point(double x = 0.0, double y = 0.0): X(x), Y(y) {}**

**~Point() {}**

**double GetX() { return X; }**

**double GetY() { return Y; }**

**private:**

**double X,Y;**

**};**

**class Triangle**

**{**

**public:**

**Triangle(Point a, Point b, Point c);**

**~Triangle() {}**

**bool IsIsosceles() const; //判断该三角形是否是等腰三角形**

**bool IsEquilateral() const; //判断该三角形是否等边三角形**

**double CalculateSide(Point p1, Point p2); //根据顶点p1和p2计算边长**

**private:**

**Point A,B,C; //三角形三个顶点**

**double AB,BC,AC; //三角形的三条边长**

**};**

**Triangle::Triangle(Point a, Point b, Point c)**

**{**

**A = a;**

**B = b;**

**C = c;**

**AB = CalculateSide(A,B);**

**AC = CalculateSide(A,C);**

**BC = CalculateSide(B,C);**

**}**

**double Triangle::CalculateSide(Point p1, Point p2)**

**{**

**Return sqrt(\_(p1.GetX()-p2.GetX())\*(p1.GetX()-p2.GetX())+(p1.GetY()-p2.GetY())\*(p1.GetY()-p2.GetY()));**

**}**

**bool Triangle::IsIsosceles() const**

**{**

**if( fabs(AB-BC) < MINNUM || fabs(AB-AC) < MINNUM || fabs(BC-AC) < MINNUM)**

**return \_ true\_\_\_\_\_\_\_\_\_\_\_;**

**else**

**return \_ false\_\_\_\_\_\_\_\_\_\_\_;**

**}**

**bool Triangle::IsEquilateral() const**

**{if (\_fabs(AB-BC)<MINNUM && fabs(AB-AC)<MINNUM && fabs(BC-AC)<MINNUM)**

**return true;**

**else**

**return false;**

**}**

**int main()**

**{**

**Point p1(2.0,0.0), p2(0.0,2.0), p3(0.0,0.0),p4(1,sqrt(3));**

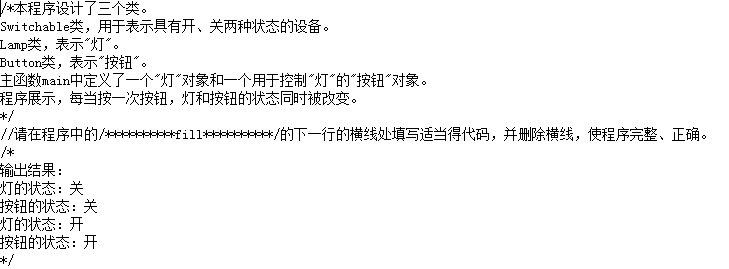
**Triangle t1(p1,p2,p3), t2(p1,p3,p4);**

**cout<<t1.IsIsosceles() << t1.IsEquilateral() << endl;**

**cout<<t2.IsIsosceles() << t2.IsEquilateral() << endl;**

**return 0;**

**}**

****

**#include<iostream>**

**using namespace std;**

**class Switchable{**

**//具有开、关两种状态的设备**

**bool is\_on;**

**//为 true 表示"开"，为 false 表示"关"**

**public:**

**Switchable(): is\_on(false){}**

**void switchOn(){ is\_on=true; }**

**//置为"开"状态**

**void switchOff(){ is\_on=false; }//置为"关"状态**

**bool isOn(){ \_\_return is\_on;\_\_\_\_\_\_\_\_\_\_\_\_\_ }**

**//返回设备状态**

**Virtual const char \*getDeviceName()\_\_\_=0\_\_\_\_;**

**//返回设备名称的纯虚函数**

**};**

**class Lamp: public Switchable{**

**public:**

**//返回设备名称，用于覆盖基类中的纯虚函数**

**const char \*getDeviceName(){ return "Lamp"; }**

**};**

**class Button{ //按钮**

**Switchable \*device; //按钮控制的设备**

**public:**

**Button(Switchable&dev):\_\_\_\_ device(&dev){} //用参数变量的地址初始化device**

**bool isOn(){ return device->isOn(); }**

**//按钮状态**

**void push(){**

**//按一下按钮改变状态**

**if(isOn()) \_\_\_\_\_\_device->switchOff()\_\_\_\_\_\_\_\_;**

**else device->switchOn();**

**}**

**};**

**int main(){**

**Lamp lamp;**

**Button button(lamp);**

**cout<<"灯的状态："<<(lamp.isOn()? "开" : "关")<<endl;**

**cout<<"按钮的状态："<<(button.isOn()? "开" : "关")<<endl;**

**button.push(); //按一下按钮**

**cout<<"灯的状态："<<(lamp.isOn()? "开" : "关")<<endl;**

**cout<<"按钮的状态："<<(button.isOn()? "开" : "关")<<endl;**

**return 0;**

**}**

**程序分析：**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**class Date{**

**public:**

**Date(int y=0, int m=0, int d=0): year(y), month(m), day(d) {}**

**~Date() {}**

**//比较date1和date2，若二者相同，则返回true，否则返回false**

**friend bool isEqual(Date date1, Date date2);**

**private:**

**int year,month,day;**

**};**

**class Plants{**

**public:**

**Plants(char \*n, Date d);**

**~Plants();**

**Date getDate();**

**char \*getName();**

**private:**

**char \*name; //植物名称**

**Date date; //该植物的开花日期**

**};**

**bool isEqual(Date date1, Date date2){**

**return date1.year==date2.year&& date1.month==date2.month &&date1.day=date2.day;**

**}**

**Plants::Plants(char \*n, Date d){**

**int length;**

**length = strlen(n);**

**name = new char[length+1] ;**

**strcpy(name,n);**

**date = d;**

**}**

**Plants::~Plants() {**

**if(!name)**

**delete []name;**

**}**

**Date Plants::getDate(){**

**return date;**

**}**

**char\* Plants::getName(){**

**return name;**

**}**

**int main(){**

**Plants plantA("桃",Date(2009,3,10));**

**Plants plantB("杏",Date(2009,4,5));**

**if(isEqual(plantA.getDate(),plantB.getDate()))**

**cout << plantA.getName() << "和" << plantB.getName() << "在同一天开花。" <<endl;**

**else**

**cout << plantA.getName() << "和" << plantB.getName() << "不在同一天开花。" <<endl;**

**return 0;**

**}**

**程序分析：**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**class Animal**

**{public:**

**Animal(char\* str="Animal")**

**{name = new char[strlen(str)+1];**

**strcpy(name,str);}**

**virtual ~Animal() { delete[] name; }**

**char\* getName() const { return name; }**

**virtual char\* getType() const { return "Animal"; }**

**virtual char\* getVoice() const { return "Voice"; }**

**private:**

**char\* name;**

**};**

**class Dog : public Animal {**

**public:**

**Dog(char\* str) : Animal(str){ }**

**char\* getType() const { return "Dog"; }**

**char\* getVoice() const { return "Woof"; }**

**};**

**class Cat : public Animal {**

**public:**

**Cat(char\* str) : Animal(str) { }**

**char\* getType() const { return "Cat"; }**

**char\* getVoice() const { return "Miaow"; }**

**};**

**void type(Animal& a)**

**{**

**cout<<a.getType()<<"named "<<a.getName();**

**}**

**void speak(Animal& a)**

**{**

**cout<<" speaks "<<a.getVoice();**

**}**

**int main()**

**{**

**Dog d("Lucky");**

**type(d);**

**speak(d);**

**cout<<endl;**

**Cat c("Daisy");**

**type(c);**

**speak(c);**

**cout<<endl;**

**return 0;**

**}**

**程序分析：**

**#include<iostream>**

**using namespace std;**

**class Base**

**{**

**public:**

**Base(int m1,int m2) {**

**mem1=m1; mem2=m2;}**

**int sum(){return mem1+mem2;}**

**private:**

**int mem1,mem2;**

**//基类的数据成员**

**};**

**// 派生类Derived从基类Base公有继承**

**class Derived: public Base**

**{**

**public:**

**//构造函数声明**

**Derived(int m1,int m2, int m3);**

**//sum函数定义，要求返回mem1、mem2和mem3之和**

**int sum(){ return Base::sum()+mem3;}**

**private:**

**int mem3;**

**//派生类本身的数据成员**

**};**

**//构造函数的类外定义,要求由m1和m2分别初始化mem1和mem2，由m3初始化mem3**

**Derived::Derived(int m1, int m2, int m3):**

**Base(m1,m2), mem3(m3){}**

**int main() {**

**Base a(4,6);**

**Derived b(10,15,20);**

**int sum=a.sum()+b.sum();**

**cout<<"sum="<<sum<<endl;**

**return 0;}**