**第三章：基本流程控制结构**

例题3.4求一元二次方程ax2+bx+c=0的解

#include<iostream>

#include<cmath>

using namespace std;

int main()

{ float a,b,c,delta;

cout<<"输入三个系数:";

cin>>a>>b>>c;

cout<<"a="<<a<<'\t'<<"b="<<b<<'\t'<<"c="<<c<<'\n';

delta=b\*b-4\*a\*c;

if(delta<0)

{cout<<"方程无实根"<<'\n' ;

}

else{

delta=sqrt(delta);

if(delta>0)

{ cout<<"方程有两个不等实根"<<'\n';

cout<<"x1="<<(-b+delta)/2/a<<'\n';

cout<<"x2="<<(-b-delta)/2/a<<'\n';

}

else{

cout<<"方程有两个相等的实根"<<'\n';

cout<<"x1=x2="<<-b/2/a<<endl;

}

}

return 0;

}

例题3.2拓展： 求n个数中最大的数

#include<iostream>

using namespace std;

int main()

{ int n,max,k,i,a,j;

cout<<"输入n个数:";

cin>>n;

cout<<'\n';

cout<<"输入第一个数k:";

cin>>k;

max=k;

a=n-1;

cout<<"输入剩下"<<a<<"个数:";

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

{ cin>>j;

if(max<j) {max=j;}

}

cout<<"最大值为"<<max;

}

例题3.6将百分制的学生成绩转换为优秀、良好、中等、及格、不及格五级制成绩

#include<iostream>

using namespace std;

int main()

{ int grade;

cout<<"输入百分制成绩:";

cin>>grade;

cout<<"五分制成绩:";

switch(grade/10)

{

case 10:

case 9:cout<<"优秀"<<'\n';

break;

case 8:cout<<"良好"<<'\n';

break;

case 7:cout<<"中等"<<'\n';

break;

case 6:cout<<"及格"<<'\n';

break;

default:cout<<"不及格"<<'\n';

}

return 0;

}

例题3.11 利用π/4=1-1/3+1/5……，最后一项绝对值小于10的负六次方

#include<iostream>

#include<cmath>

using namespace std;

int main()

{ int i=1,sign=1; //设置符号标志

float s=0,t=1; //累加和为浮点数

while(fabs(t)>1e-6) //t的绝对值大于10的负6次方

{ t=sign\*1.0/i; //通项的表示

sign=-sign; //符号取反

i=i+2; //分母进行迭代

s=s+t; //计算累加和

}

cout<<"pi="<<4\*s<<'\n';

return 0;

}

例题3.13 用迭代法求x=sqrt a的近似值(拓）

#include<iostream>

#include<cmath>

using namespace std;

int main()

{ float x0,x1,a;

cout<<"请输入一个正数:";

cin>>a;

if(a<0) {cout<<"a不能开平方！"<<'\n';}

else { x1=a/2;

do{ x0=x1;

x1=(x0+a/x0)/2;

}

while(fabs(x1-x0)>1e-5);

cout<<"a的平方根为："<<x1<<'\n';

}

return 0;

}

**打印图案专题**

例题3.15打印图案

#include<iostream>

using namespace std;

int main()

{ int i,j,k;

for(i=1;i<=4;i++) //控制上面四行

{ for(j=1;j<=4-i;j++) //控制\*前的空格数

{cout<<' ';} //输出一个空格

for(k=1;k<=2\*i-1;k++) //控制\*的个数

{cout<<'\*'; } //输出完一行后换行

cout<<'\n';

}

for(i=1;i<=3;i++) //控制下面三行

{ for(j=1;j<=i;j++) //控制\*前的空格数

{cout<<' '; } //输出一个空格

for(k=1;k<=2\*(3-i)+1;k++) //控制\*的个数

{cout<<'\*';}

cout<<'\n';

}

return 0;

}

**拓展：打印九九乘法表**

**九九乘法表右三角（上多下少）**

#include <iostream>

#include<iomanip>

using namespace std;

int main()

{

int m=0, n=0;

for (m=1; m<10; m++) //第一行

{ cout<<setw(4)<<m; } //打印表头

cout<<endl;

for (m=1; m<10; m++) //第二行字符（三个空格）对齐

{cout<<" -";}

cout<<endl;

for (m=1; m<10; m++) //m为行，n为列

{

for (n=1; n<=10-m; n++)

{

cout<<setw(4)<<m \* n;

}

cout<<endl;

}

return 0;

}

**九九乘法表右三角（上少下多）**

#include <iostream>

#include<iomanip>

using namespace std;

int main()

{

int m=0, n=0;

for (m=1; m<10; m++)

{ cout<<setw(4)<<m; } //打印表头

cout<<endl;

for (m=1; m<10; m++)

{cout<<" -";}

for (m=1; m<10; m++)

{

for (n=1; n<m; n++)

{

cout<<setw(4)<<m \* n;

}

cout<<endl;

}

return 0;

}

**九九乘法表左三角 （上多下少）**

#include <iostream>

#include<iomanip>

using namespace std;

int main()

{

int m=0, n=0;

for (m=1; m<10; m++) //第一行

{ cout<<setw(4)<<m; } //打印表头

cout<<endl;

for (m=1; m<10; m++) //第二行字符（三个空格）对齐

{cout<<" -";}

cout<<endl;

for (m=1; m<10; m++) //m为行，n为列

{

for (n=1; n<10; n++)

{ if(n<m)

{cout<<" ";}

else cout<<setw(4)<<m \* n;

}

cout<<'\n';

}

return 0;

}

**九九乘法表左三角 （上少下多）**

#include <iostream>

#include<iomanip>

using namespace std;

int main()

{

int m=0, n=0;

for (m=1; m<10; m++) //第一行

{ cout<<setw(4)<<m; } //打印表头

cout<<endl;

for (m=1; m<10; m++) //第二行字符（三个空格）对齐

{cout<<" -";}

cout<<endl;

for (m=1; m<10; m++) //m为行，n为列

{

for (n=1; n<10; n++)

{ if(n<10-m)

{cout<<" "; //不输出m\*n时要打空格

}

else cout<<setw(4)<<m \* n;

}

cout<<endl;

}

return 0;

}

**打印菱形**

#include<iostream>

using namespace std;

int main()

{ int m,i,j; //m表示行数，i用于输出\*前的空格，j用于输出\*

for(m=1;m<=2;m++)

{ for(i=1;i<=3-m;i++) // 行数：1 2 3 4 5

{ cout<<" ";} // 空格：2 1 0 1 2

for(j=1;j<=m;j++) // \* ：1 2 2 2 1

{ cout<<"\* ";}

cout<<'\n';

}

if(m=3) {cout<<"\* \*";}

cout<<'\n';

for(m=4;m<=5;m++)

{ for(i=1;i<=m-3;i++)

{cout<<" ";}

for(j=1;j<=6-m;j++)

{cout<<"\* ";}

cout<<'\n';

}

}

**打印平行四边形**

#include<iostream>

using namespace std;

int main()

{ int m,i,j;

for(m=1;m<=4;m++)

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

{cout<<" ";}

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

{cout<<"\*";}

cout<<'\n';

}

}

**打印沙漏**

#include<iostream>

using namespace std;

int main()

{ int i,j,k;

for(i=1;i<=3;i++)

{ for(j=1;j<=i-1;j++) //控制\*前的空格数 1 2 3 4 5

{cout<<' ';} //空格 0 1 2 1 0

for(k=1;k<=7-2\*i;k++) //控制\*的个数 \* 5 3 1 3 5

{ cout<<'\*';}

cout<<'\n';

}

for(i=4;i<=5;i++)

{ for(j=1;j<=5-i;j++)

{cout<<' '; }

for(k=1;k<=2\*i-5;k++)

{cout<<'\*';}

cout<<'\n';

}

return 0;

}

例题3.16 求三个整数的最小公倍数和最大公约数

#include<iostream>

using namespace std;

int main()

{ int a,b,c,x,y;

cout<<"请输入三个整数:";

cin>>a>>b>>c;

for(x=a;x<a\*b\*c;x++) //x从小到大逐一测试

{ if(x%a==0&&x%b==0&&x%c==0) //公倍数的选择条件

break;

}

for(y=a;y>0;y--)

{ if(a%y==0&&b%y==0&&c%y==0)

break;

}

cout<<a<<","<<b<<","<<c<<"的最小公倍数:"<<x<<endl;

cout<<a<<","<<b<<","<<c<<"的最大公约数:"<<y<<endl;

return 0;

}

例题3.18 输出10~20之间不是3的倍数的数

#include<iostream>

using namespace std;

int main()

{ int i;

for(i=10;i<=20;i++)

{ if(i%3==0) continue;

{cout<<i<<'\t';}

}

cout<<'\n';

return 0;

}

例3.20 用公式e=1+1/1！……+1/n！+……求e

#include<iostream>

using namespace std;

int main()

{ double e=1.0,t=1.0;

int n=1;

while(t>=1e-7)

{ t=t/n;

e=e+t;

n++;

}

cout<<"e="<<e<<endl;

return 0;

}

**例题3.21 利用随机数的函数库rand设计一个自动出题程序，运算时两个操作数的取值范围为0~9；共出10题，每题10分，最后给出总分。**

#include<iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{ int i,a,b,sum=0;

int op,c,d;

srand(time(NULL));

for(i=1;i<=10;i++)

{ a=rand()%10;

b=rand()%10;

op=rand()%3; //共有三种运算，分别用0,1,2表示

switch(op)

{case 0:cout<<a<<'+'<<b<<'=';

c=a+b; break;

case 1:cout<<a<<'-'<<b<<'=';

c=a-b; break;

case 2:cout<<a<<'\*'<<b<<'=';

c=a\*b;

}

cin>>d; //接收用户从键盘上输入答案

if(d==c)

{ cout<<"正确！"<<'\n';

sum=sum+10;

}

else cout<<"错误"<<'\n';

}

cout<<"10题中答对:"<<sum/10<<"题,"<<'\t'<<"得分:"<<sum<<endl;

return 0;

}

**课后习题**

编程2 从键盘输入一个不多于5位的正整数，能输出它的位数并输出它的各位数字之和

#include<iostream>

using namespace std;

int main()

{ int a,b,c,d,e,x,i,j=0,k,t=0,z=0;

cout<<"请输入一个数x:";

cin>>x;

if(x>0&&x<=99999)

{

k=x;

t=x;

do

{ t=k%10;

i=k/10; //eg: x=1023 t1=3 i1=102 k1=102 z1=3 j1=1

k=k/10; // t2=2 i2=10 k2=10 z2=5 j2=2

z=z+t; // t3=0 i3=1 k3=1 z3=5 j3=3

j=j++; // t4=1 i4=0 k4=0 z4=6 j4=4

//所以1023为四位数，各个位数之和为6

}while(i>0);

cout<<"x为"<<j<<"位数";

cout<<'\n';

cout<<"x的各个位数之和为:"<<z;

}

else cout<<"x不合题意";

return 0;

}

编程6：求水仙花数（一个三位数，各位数字的立方和等于其本身）

#include<iostream>

using namespace std;

int main()

{ int a,b,c,i=100;

cout << "水仙花数为:" << endl;

for(i=100;i<=999;i++)

{ a=i/100;

b=i%100/10;

c=i%10;

if(i==a\*a\*a+b\*b\*b+c\*c\*c)

cout<<i<<'\n';

}

return 0;

}

**编程8 辗转相除法求两数最大公约数**

#include<iostream>

using namespace std;

int main()

{ int m,n,r,t;

cout << "输入两个正整数m和n:";

cin>>m>>n;

if(m<n) {t=m; m=n; n=t;}

while(r!=0)

{ r=m%n;

m=n;

n=r;

}

cout<<"m,n的最大公约数为："<<m<<endl;

return 0;

}

**第四章：函数**

例题4.2 无参函数的例子

#include<iostream>

using namespace std;

void message(void)

{ cout<<"1.注册新用户\n";

cout<<"2.用户登录\n";

cout<<"3.取消\n";

cout<<"请输入您的选择:";

}

int main()

{ char ch;

message();

ch=cin.get();

cout<<endl<<"您选择了第"<<ch<<"项"<<endl;

return 0;

}

例题4.3 求1！+2！+……+n！（函数）

#include<iostream>

using namespace std;

float fact(int m) //定义一个fact函数用于求n！

{ float product=1;

if(m<0) return -1;

else if(m==0) return 1;

while(m>=1)

{ product=product\*m;

m--;

}

return product;

}

int main()

{ int i,n;

float s=0;

cout<<"请输入一个正整数:";

cin>>n;

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

{ s=s+fact(i);

}

cout<<"s="<<s<<'\n';

return 0;

}

例题4.4 求三个数的最大公约数（函数）

#include<iostream>

using namespace std;

int gys(int a,int b,int c) //三个数的最大公约数就是可以被这三个数整除的最大的数

{ int i;

for(i=a;i>=1;i--) //由大到小循环查找公约数

{ if(a%i==0&&b%i==0&&c%i==0) //判断公约数的条件

break; //满足条件，退出循环

}

return i;

}

int main()

{ int a,b,c;

cout<<"请输入三个数:";

cin>>a>>b>>c;

cout<<a<<","<<b<<","<<c<<"的最大公约数是:"<<gys(a,b,c); //调用函数

return 0;

}

例题4.7 **函数的嵌套调用** 编程求f（k，n）=1^k+2^k+……n^k (n,k从键盘上输入)

#include<iostream>

using namespace std;

int pow(int m ,int k)

{ int p=1;

for(int i=1;i<=k;i++)

{ p=p\*m;}

return p;

}

int sum(int n,int k)

{ int i,s=0;

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

{ s=s+pow(i,k);

}

return s;

}

int main()

{ int n,k;

cout<<"请输入n和k的值:";

cin>>n>>k;

cout<<"1^k+2^k+3^k+……+"<<n<<"^k="<<sum(n,k)<<endl;

return 0;

}

例题4.8 **函数的递归调用：**用递归方法计算n！

#include<iostream>

using namespace std;

float fact(int n)

{ float p;

if(n==0||n==1) p=1;

else p=n\*fact(n-1); //求n！的函数为fact（n）

return p;

}

int main()

{ int n;

cout<<"n=";

cin>>n;

cout<<n<<"!="<<fact(n)<<'\n';

return 0;

}

例题4.9 **函数的递归调用**：分析程序运行结果

#include<iostream>

using namespace std;

void recu(char c)

{ cout<<c;

if(c<'3') recu(c+1);

cout<<c;

}

int main()

{ recu('0');

return 0;

}

答案：01233210

例题4.12**函数重载：**利用重载函数求两个数的加法

#include<iostream>

using namespace std;

int sum(int a,int b) //A

{ return a+b;}

double sum(double a,double b) //B

{ return a+b;}

int main()

{ cout << "3+5=" << sum(3,5)<<endl; //实参为整数，调用A的sum函数

cout << "3.5+8.7=" << sum(3.5,8.7)<<endl; //实参为浮点数，调用B的sum函数

return 0;

}

例题4.13 **带有默认参数的函数：**利用带有默认参数的函数求长方形面积

#include<iostream>

using namespace std;

int area(int length=4,int width=2)

{ return length\*width;}

int main()

{ int a=8,b=6;

cout << area(a,b)<< endl; //A 相当于调用area(8,6)

cout<<area(a)<<endl; //B 相当于调用area(8,2)

cout<<area()<<endl; //C 相当于调用area(4,2)

return 0;

}

**课后习题**

1. **回文数**

**任意一个正整数与其反序数相加得到一个新的整数，重复该步骤即可得到回文数**

#include<iostream>

using namespace std;

int invert(int x) //求一个整数的反序数

{ int s;

for(s=0;x>0;x=x/10)

{s=s\*10+x%10;}

return s;

}

int main()

{ int n,c=0;

cout<<"input a number:";

cin>>n;

while(n<=0)

{ cout<<"input a number:";

cin>>n;

}

n=n+invert(n);

c++;

while(n!=invert(n))

{ n=n+invert(n);

c++;

}

cout<<n<<",count="<<c<<endl; //n即为最终的回文数

return 0;

}

编程3：求1000-2000的回文数(具体方法）

#include<iostream>

using namespace std;

int fun( int x)

{ int prime=1,a,b,c,d;

a=x/1000;

b=x/100%10;

c=x/10%10;

d=x%10;

if(a==d&&b==c) prime=0;

return prime;

}

int main()

{ int i,j;

cout << "1000到2000的回文数为:" << endl;

for(i=1000;i<=2000;i++)

{ j= fun(i);

if(j==0) cout <<i<<'\n';

}

return 0;

}

拓：P167完善1 求11-999之间所有满足m、m×m、m×m×m都为回文数的整数m

#include<iostream>

using namespace std;

int f(int n)

{ int i=0,j=0,a[10];

while(n!=0) **//求n的反序数**

{ a[j++]=n%10; eg：123->j=0 a[0]=3 j=1 a[1]=2 j=2 a[2]=1 j=3时退出循环

n=n/10;

}

j--; //j=3-1=2

while(i<j)

{ if(a[i]==a[j]) i++,j--; //a[0]=a[2]? a[1]=a[1]?

else return 0;

}

return 1;

}

int main()

{ int m;

for(m=11;m<1000;m++)

if(f(m)&&f(m\*m)&&f(m\*m\*m))

cout<<"m="<<m<<'\t'<<"m\*m="<<m\*m<<'\t'<<"m\*m\*m="<<m\*m\*m<<endl;

return 0;

}

编程2：求100-200的素数，每输出5个换行（函数）

#include<iostream>

using namespace std;

int fun(int a)

{ int i,prime=1;

for(i=2;i<a;i++)

{ if(a%i==0) prime=0;

}

return prime;

}

int main()

{ int j,x,num=0;

for(j=100;j<=200;j++)

{ x=fun(j);

if(x==1)

{ cout<<j<<'\t';

num++;

if(num%5==0) //只有输出素数才需要判断是否需要换行

{cout<<'\n';}

}

}

return 0;

}

编程5：递归函数求hermite多项式的前n项值

#include<iostream>

using namespace std;

int fun(int n,int x)

{ int p;

if(n==0) p=1;

if(n==1) p=2\*x;

if(n>1)

{p=2\*x\*fun(n-1,x)-2\*(n-1)\*fun(n-2,x);}

return p;

}

int main()

{ int n,x;

cout << "输入n,x:";

cin >> n>>x;

for(n;n>=0;n--)

{

cout << "H"<<n<<"("<<x<<")"<<"=:"<<fun(n,x)<<'\n';

}

return 0;

}

**编程6：输出一个数的正序与反序**

**本质：反序输出：先输出余数再递归**

**正序输出：先递归在输出余数**

#include<iostream>

using namespace std;

void descend(int n)

{if(n==0)

return;

cout<<n%10;

descend(n/10); //反序

}

void ascend(int n)

{if(n==0)

return;

ascend(n/10); //正序

cout<<n%10;

}

int main()

{int n,a;

cout<<"请输入一个整数:"<<'\n';

cin>>n;

cout<<"正序请输入1"<<'\n';

cout<<"反序请输入2"<<'\n';

cin>>a;

if(a==1)

ascend(n);

cout<<'\n';

if(a==2)

descend(n);

return 0;

}

**第六章：数组**

一维数组

编程6.3：用冒泡法对10个数由小到大排序

本质：**外层n-1次循环 内层n-i-1次循环（在内层中相邻两项比较大小）**

#include<iostream>

using namespace std;

int main()

{ int a[10];

int i,j,t;

cout<<"Input 10 number:\n";

for(i=0;i<10;i++)

{ cin>>a[i]; //输入数组

}

for(i=0;i<=10-2;i++)

{ for(j=0;j<=10-i-2;j++)

{ if(a[j]>a[j+1])

{ t=a[j]; //交换两个数

a[j]=a[j+1];

a[j+1]=t;

}

}

}

for(i=0;i<10;i++)

{ cout<<a[i]<<'\t';

}

return 0;

}

编程6.4：将数组中指定的数据删除

**本质：后面的往前覆盖**

#include<iostream>

using namespace std;

int main()

{ int a[6]={1,3,5,7,9,11},i,j,x;

cout<<"原数组是:\n";

for(i=0;i<=5;i++)

{ cout<<a[i]<<'\t';}

cout<<'\n';

cout<<"请输入想要删除的数据:";

cin>>x;

for(i=0;i<=5;i++)

{ if(a[i]==x) //找删除点

{ **for(j=i;j<=4;j++)**

{ a[j]=a[j+1]; } //从删除点后的数据开始依次向前覆盖

break; //覆盖完数据后退出循环

}

}

if(i==6) //在i=0~5都没找到符合条件的数

cout<<"数组中没有要删除的数据\n";

else

{ cout<<"删除数据"<<x<<"后的数组为:\n";

for(i=0;i<=4;i++)

{ cout<<a[i]<<'\t';

}

}

return 0;

}

二维数组

例题6.6 将一个3乘3的二维数组行和列元素互换

**法一：用新的数组b存放**

#include<iostream>

using namespace std;

int main()

{ int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};

int b[3][3],i,j;

cout<<"数组a:\n";

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

{ for(j=0;j<3;j++)

{ cout<<a[i][j]<<'\t';

}

cout<<'\n';

}

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

{ for(j=0;j<3;j++)

b[j][i]=a[i][j]; //将a数组的i行j列 赋给b数组的j行i列

}

cout<<"数组b:\n";

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

{ for(j=0;j<3;j++)

{ cout<<b[i][j]<<'\t';}

cout<<'\n';

}

return 0;

}

**法二：遍历对角线上方的元素进行交换**

#include<iostream>

using namespace std;

int main()

{ int i,j,t;

int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};

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

{  **for(j=i;j<3;j++) //遍历对角线上方的元素进行互换**

{ t=a[i][j];

a[i][j]=a[j][i]; //行列互换

a[j][i]=t;

}

}

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

{ for(j=0;j<3;j++)

{cout<<a[i][j];}

cout<<endl;

}

return 0;

}

**数组的元素作为函数的实参**

例题6.7 求第三个数据系列c是数组a，b中对应数的最大公约数

#include<iostream>

using namespace std;

int gys(int m,int n) //求两个数的最大公约数，形参是变量

{ int r,t;

if(m<n) //算法要求m>n

{ t=m;

m=n;

n=t;

}

do

{ r=m%n;

m=n;

n=r;

} while(r>0);

return m;

}

int main()

{ int a[8]={26,1007,956,705,574,371,416,517};

int b[8]={994,631,772,201,262,763,1000,781};

int c[8],i;

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

**{ c[i]=gys(a[i],b[i]); } //对应元素的公约数，实参是数组元素**

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

cout<<c[i]<<'\t';

cout<<endl;

return 0;

}

**例题6.9 用选择法对10个数由小到大排序**

#include<iostream>

using namespace std;

void sort(int a[],int n) //选择法排序

{ int i,j,min,t;

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

**{ min=i; //设置min的初值**

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

**{ if(a[min]>a[j]) //循环找出最小值的序号赋给min （先找完一轮）**

**min=j;**

**}**

**t=a[i]; //交换a[i]和a[min]**

**a[i]=a[min];**

**a[min]=t;**

**}**

}

int main()

{ int b[]={9,9,4,0,2,5,8,2},i,n;

n=sizeof(b)/sizeof(b[0]); //sizeof用于求字节数

sort(b,n); //用选择法排序函数

cout<<"排序后的数组如下:"<<'\n';

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

{ cout<<b[i]<<'\t';

}

cout<<endl;

return 0;

}

例题6.10 删除一个数组中相同的数据，使之只剩下1个

#include<iostream>

using namespace std;

int del(int a[],int n) //删除相同数据(比较相邻的数是否相等)

{ int i,j;

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

**{ if(a[i]==a[i+1]) //判断相邻的数是否相等**

**{ for(j=i;j<n-1;j++) //若相等，则删除前一个数**

**{ a[j]=a[j+1];}**

**n--; //删除后元素个数减少**

**i--; //判断点前移，继续判断i之后是否有相等的数**

**}**

}

return n; //将删除后的数据个数返回

}

int main()

{ int b[]={1,1,1,2,2,2,3,3,4,4,4,5,5,6,6,6,6},i,n;

n=sizeof(b)/sizeof(b[0]);

n=del(b,n); //调用del函数，返回删除后数组元素的个数

cout<<"删除相同数据后的数组为:\n";

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

{ cout<<b[i]<<'\t';}

cout<<endl;

return 0;

}

例题6.15 **字符数组的复制与拼接：**分析下列程序输出结果

#include<iostream>

#include<cstring>

using namespace std;

int main()

{ char str[10]="12345",str1[10]="ABCDE",str2[]="abcde";

strcpy(str1+4,str2+3);

strcat(str,str1+2);

cout<<str<<endl;

return 0;

}

答案：12345CDde

例题6.16 **字符数组的复制与比较大小：**输入三个字符串，按升序排序后输出

#include<iostream>

#include<cstring>

using namespace std;

void swap(char a[],char b[]) //实现两个字符串的交换

{ char t[80];

strcpy(t,a); strcpy(a,b); strcpy(b,t);

}

int main()

{ char s1[80],s2[80],s3[80];

cout<<"输入三个字符串:\n";

cin.getline(s1,80); //输入字符串

cin.getline(s2,80);

cin.getline(s3,80);

if(strcmp(s1,s2)>0) swap(s1,s2); //判断字符串大小

if(strcmp(s1,s3)>0) swap(s1,s3);

if(strcmp(s2,s3)>0) swap(s2,s3);

cout<<"排序后的结果为:\n";

cout<<s1<<endl<<s2<<endl<<s3<<endl;

return 0;

}

例题6.17 **字符数组的遍历和复制：**将字符串中的所有数字字符删除

#include<iostream>

#include<cstring>

using namespace std;

void del(char s[])

{ **int i=0;**

**while(s[i]) //遍历数组**

{ if(s[i]>='0'&&s[i]<='9') //当前字符为数字字符

{ strcpy(s+i,s+i+1); //将当前字符后面的字符串覆盖到当前字符

i--; // (判断点前移)

}

i++;

}

}

int main()

{ char str[80];

cout<<"请输入一个字符串:"<<'\n';

cin.getline(str,80);

del(str);

cout<<str<<endl;

return 0;

}

例题6.18 输入一个字符串和子串，将字符串中的**所有子串删除**

#include<iostream>

#include<cstring>

using namespace std;

int del(char s[],char sub[])

{ int i=0,j,k,flag=0;

while(s[i]) //遍历数组

{ j=i;k=0;

while(s[j]==sub[k]&&s[j]!='\0') //遍历子串比较

{ j++;

k++;

}

if(sub[k]=='\0')

{ flag=1; //找到子串后设置标记位

strcpy(s+i,s+j); //删除子串

i--;

}

i++;

}

if(flag==1) return 1; //若找到子串返回1

else return 0;

}

int main()

{ char str[80],substr[10];

int n;

cout<<"请输入一个字符串:";

cin.getline(str,80);

cout<<"请输入一个子串:";

cin.getline(substr,10);

n=del(str,substr);

if(n==1) cout<<"删除子串后的字符串为:"<<str;

else cout<<str<<"中未找到"<<substr<<"子串"<<endl;

return 0;

}

例题6.19 **字符数组中的插入：**将字符串中所有数字字符子串前插入一个符号'-'

#include<iostream>

#include<cstring>

using namespace std;

void insert(char a[],char b[])

{ int i,j,flag=0;

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

{ if(!(a[i]>='0'&&a[i]<='9')) //不是数字字符，将字符复制到数组b

{ b[j++]=a[i];

flag=1;

}

else //数字字符

{ if(flag==1)

{ b[j++]='-'; //第一个数字字符插入'-'

flag=0;

}

b[j++]=a[i]; //其余数字字符依次复制到数组b

}

}

b[j]='\0';

}

int main()

{ char str1[80],str2[80];

cout<<"请输入一个字符串:";

cin.getline(str1,80);

insert(str1,str2);

cout<<str2<<endl;

return 0;

}

**课后习题**

完善2 **可逆素数：一个数的反序数是素数**.在【m，n】区间查找所有可逆素数并依次保存到a指向的数组

#include<iostream>

#include<cmath>

using namespace std;

int p(int n)

{ int i,j=sqrt((double)n);

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

if(n%i==0) return 0;

else return 1;

}

**int convert(int n) //求一个数的反序数**

**{ int m=0;**

**while(n>0)**

**{m=m\*10+n%10;**

**n=n/10;**

**}**

**return m;**

}

int f(int m,int n,int a[])

{ int i,j=0;

for(i=m;i<=n;i++)

if(p(i)&&p(convert(i))) a[j++]=i;

else return j;

}

int main()

{ int i,n,a[50];

n=f(50,150,a);

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

cout<<a[i]<<'\t';

return 0;

}

完善3 itoa16函数将int型**整数a转换成16进制数字字符串**，并保存到p指向的字符数组中

#include<iostream>

using namespace std;

void itoa16(int a,char p[])

{ int i=0,j=0,k,r,t[10];

if(a<0)

{ p[j]='-';

j++;

a=-a;

}

p[j]='0';

j++;

p[j]='x';

j++;

**while(a>0)**

**{ r=a%16;**

**if(r<10) t[i]=r+'0';**

**else t[i]=r-10+'A';**

**a=a/16;**

**i++;**

**}**

for(k=i-1;k>=0;k--,j++) p[j]=t[k];

p[j]='\0';

}

int main()

{ int a=127;

char b[10];

itoa16(a,b);

cout<<b<<endl; //127=16\*7+15=0x7F

return 0;

}

编程2 输出杨辉三角行（8行）

#include<iostream>

using namespace std;

int main()

{ int a[8][8]={{1}}; //先给第一个元素赋值为1,其他元素均为0

int b,c,d,i,j,k,z;

for(b=0;b<8;b++)

a[b][0]=1; //第一列的元素全部赋值为1

for(c=0,d=0;c<8;c++,d++)

a[c][d]=1; //行数等于列数的元素全部赋值为1

{for(i=2;i<8;i++) //从第二行开始

{for(j=1;j<i+1;j++) //列数大于行数时才需要利用下列公式给矩阵元素赋值

**a[i][j]=a[i-1][j-1]+a[i-1][j]**; //eg.a[2][1]=a[1][0]+a[1][1]

}

}

for(k=0;k<8;k++)

{ for(z=0;z<=k;z++) //当列数<行数时才输出矩阵

cout<<a[k][z]<<'\t';

cout<<'\n';

}

return 0;

}

编程4 删除一个字符串中的所有空格并输出

#include<iostream>

using namespace std;

int main()

{ int i,j;

char ch[100];

cout<<"输入字符串:";

cin.getline(ch,100);

for(i=0;i<100;i++)

{ if(ch[i]==32) //空格的ASCII码为32

{ for(j=i;j<99;j++)

{ch[j]=ch[j+1];}

i--; //继续判断是否有相同的数据

}

}

cout << "删除空格后的数组为:" <<ch;

}

编程5 反序输出一串字符

#include<iostream>

#include<cstring>

using namespace std;

int main()

{ int i,j,n,t;

char ch[100];

cout << "输入一个字符串:";

cin.getline(ch,100);

n=strlen(ch); //计算ch的字节数

**for(i=0;i<n/2;i++)**  //小于n/2的元素与大于n/2的元素交换

{ t=ch[i];

ch[i]=ch[n-1-i];

ch[n-1-i]=t;

}

cout << "反序输出为:" <<ch;

}

编程6 将10进制数转化为16进制

#include <iostream>

#include<cstring>

using namespace std;

void change(int n, char str[])

{

int i=0,j;

int x;

char ch;

while(n)

{

x=n%16;

if(x<10)

str[i]=x+'0';

else

str[i]=x-10+'A';

n=n/16;

i++;

}

str[i]='\0';

for(i=0,j=strlen(str)-1;i<j;i++,j--)

{

ch=str[i];

str[i]=str[j];

str[j]=ch;

}

}

int main()

{

int n;

char str[80];

cout<<"请输入一个十进制整数: ";

cin>>n;

change(n,str);

cout<<n<<"的十六进制数的形式为: "<<str<<"H"<<endl;

return 0;

}

编程7 编写函数将b的字符串复制给数组a（不用strcpy）

#include <iostream>

using namespace std;

void strcpy(char a[],char b[])

{

**int i=0,j=0;**

**while(a[i++]=b[j++]);**

}

int main()

{

char a[80],b[80];

cout<<"请输入一个字符串: ";

cin.getline (b,80);

strcpy(a,b);

cout<<"字符串b为: "<<b<<endl;

cout<<"字符串a为: "<<a<<endl;

return 0;

}

编程8 将b中的字符串拼接到数组a中的字符串的后面，构成一个字符串（不用strcat）

#include <iostream>

using namespace std;

int strcat(char a[],char b[])

{

**int i=0,j=0;**

**while(a[i])**

**i++;**

**while(a[i++]=b[j++]);**

}

int main()

{

char a[80],b[80];

cout<<"请输入两个字符串: ";

cin.getline (a,80);

cin.getline(b,80);

cout<<"字符串a为: "<<a<<endl;

cout<<"字符串b为: "<<b<<endl;

strcat(a,b);

cout<<"字符串a+b为:"<<a<<endl;

return 0;

}

**编程10 从键盘输入一个整型数，把这个整型数的各位数按降序输出**

#include <iostream>

using namespace std;

void sort(int a[],int n) //选择法排序（降序）

{  **int i,j,max;**

**int t;**

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

**{ max=i;**

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

**{**

**if(a[max]<a[j])**

**max=j;**

**}**

**t=a[i];**

**a[i]=a[max];**

**a[max]=t;**

**}**

**}**

int main()

{

int x;

int i=0,n;

int a[100];

cout<<"请输入一个整数: ";

cin>>x;

**while(x) //用数组存放一个整数的各位数字（反序排列）**

**{**

**a[i]=x%10;**

**x=x/10;**

**i++;**

**}**

n=i; //数组元素的个数

sort(a,n);

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

cout<<a[i];

cout<<endl;

return 0;

}

**第七章：指针**

例题7.4 用指针变量作函数参数

#include<iostream>

using namespace std;

void change(int \*p1,int \*p2)

{ int t;

t=\*p1; \*p1=\*p2; \*p2=t;

}

int main()

{ int a,b;

cout<<"请输入两个数:";

cin>>a>>b;

if(a<b) change(&a,&b);

cout<<"a="<<a<<'\t'<<"b="<<b<<endl;

return 0;

}

例题7.6 **指针与一维数组：**用指针将数组元素反向排序

#include<iostream>

using namespace std;

**void inverse(int \*p,int n)**

**{ int \*q=p+n-1; //指针变量q指向数组最后一个元素**

**int t;**

**while(p<q)**

**{ t=\*p;**

**\*p=\*q; //交换\*p和\*q**

**\*q=t;**

**p++; q--; //移动指针变量**

**}**

**}**

int main()

{ int a[10]={1,2,3,4,5,6,7,8,9,10};

inverse(a,10);

for(int i=0;i<10;i++)

cout<<a[i]<<'\t';

return 0;

}

例题7.7 **指针版：**将字符数组a的字符串复制到b中

#include<iostream>

using namespace std;

int main()

{ char a[100]="Hello,world",b[100]="khfeoijefifeoief";

char \*p1=a,\*p2=b;

**while(\*p1)**

**{ \*p2++=\*p1++;}**

**\*p2='\0';**

cout<<b<<endl;

return 0;

}

例题7.8 用指针删除字符串中的指定字符

#include<iostream>

#include<cstring>

using namespace std;

void delchar(char \*p,char c)

{ while(\*p)

{ if(\*p==c)

{ strcpy(p,p+1);

p--; //同一位置判断两次

}

p++;

}

}

int main()

{ char a[100]="student",c='t';

delchar(a,c); //删除字符't'

cout<<a<<endl;

return 0;

}

**拓展联想：联想6.10与7.8（删除相同数据：一个不留、任意数组）**

#include<iostream>

using namespace std;

int del(int a[],int n,int c)

{ **int \*p,\*q,sum=0;**

**p=&a[0];**

**for(p=&a[0];p<&a[n-1];p++,sum++)**

**{ if(\*p==c)**

**{ for(q=p;q<&a[n-1];q++) //定义新的指针q指向当前p指向的位置，后面的往前覆盖**

**{\*q=\*(q+1);}**

**n--;**

**p--;**

**}**

**}**

**return n;**

}

int main()

{ int b[]={9,1,2,2,2,3,3,4,4,4,6,6,5,5,6,4,6,7,9,80},i,n,t;

n=sizeof(b)/sizeof(b[0]);

t=del(b,n,6); //调用del函数，返回删除后数组元素的个数

cout<<"删除数据6后的数组为:\n";

for(i=0;i<t;i++)

{ cout<<b[i]<<'\t';}

cout<<endl;

return 0;

}

例题7.9 **函数指针变量：**通过函数指针变量调用函数

#include<iostream>

using namespace std;

int add( int a,int b)

{return a+b; }

int main()

{ int x=2,y=3,t;

int (\*p)(int,int); //定义函数指针

p=add; //将add函数入口地址赋给函数指针

t=p(x,y); //通过函数指针调用add函数

cout<<t<<endl;

return 0;

}

例题7.10 **函数指针作为函数参数：**用二分法求下列方程的根

#include<iostream>

#include<cmath>

using namespace std;

float f1(float x)

{return x\*x\*x+x\*x-3\*x+1;} //f(1)x

float f2(float x)

{return x\*x-2\*x-8;} //f(2)x

float f3(float x)

{return x\*x\*x+2\*x\*x+2\*x+1;} //f(3)x

float divide(float(\*p)(float),float x1,float x2) //二分法计算方程的解

{ float x0;

do

{ x0=(x1+x2)/2;

if(p(x1)\*p(x0)>0) //若p(x1)\*p(x0)>0说明解在x2与x0之间

x1=x0;

else //若p(x1)\*p(x0)<0说明解在x1与x0之间

x2=x0;

}while(fabs(p(x0))>1e-6);

return x0;

}

int main()

{ cout<<"f1方程的解为:"<<divide(f1,-2,2)<<endl;

cout<<"f2方程的解为:"<<divide(f2,-3,3)<<endl;

cout<<"f3方程的解为:"<<divide(f3,-2,3)<<endl;

return 0;

}

例题7.13 指针与二维数组 **二维数组的行指针与列指针**：

3个学生，4门课，用指针计算平均分数和输出不及格分数

#include<iostream>

using namespace std;

float average(int (\*p)[4],int n) //用行指针变量作形参

{ float aver=0;

for(int i=0;i<n;i++,p++) //行指针变量自加，指向下一行

for(int j=0;j<4;j++)

aver=aver+(\*p)[j];

aver=aver/n/4; //n代表人数

return aver;

}

void output(int \*p,int n)

{ for(int i=0;i<n;i++,p++) //列指针变量自加，指向下一个元素

if(\*p<60)

cout<<\*p<<'\t';

cout<<endl;

}

int main()

{ int a[3][4]={{80,90,80,70},{60,50,70,75},{55,80,75,75}};

float aver;

aver=average(a,3);

output(a[0],12);

cout<<"average="<<aver<<endl;

return 0;

}

例题7.15 **指针数组：**将若干字符串按字母顺序由小到大排序

#include<iostream>

#include<cstring>

using namespace std;

void sort(char \*str[],int n)

{ char\*p;

int i,j;

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

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

if(strcmp(str[j],str[j+1])>0)

{ p=str[j];

str[j]=str[j+1];

str[j+1]=p;

}

}

int main()

{ char \*week[]={"Monday","Tuesday","Wednesday","Thursday","Friday"};

sort(week,5); //对字符串排序

for(int i=0;i<5;i++)

cout<<week[i]<<'\n';

return 0;

}

例题7.16 **指向指针的指针：**分析指向指针的指针运行结果

#include<iostream>

using namespace std;

int main()

{ int a[9]={1,2,3,4,5,6,7,8,9};

int \*b[]={&a[0],&a[3],&a[6]};

int \*\*p=b;

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

cout<<p[i][i]<<'\t'; //p[1][1]=\*(p[1]+1)=\*(\*(p+1)+1)=\*(b[1]+1)=\*(a[3]+1)=a[4]

return 0;

}

例题7.18 **指针的动态存储分配：**根据学生人数动态分配存储空间，计算学生平均成绩

#include<iostream>

using namespace std;

int main()

{ int \*p,n;

double s=0;

cout<<"请输入学生数:";

cin>>n;

p=new int[n]; //根据n的数目动态分配存储空间，空间的首地址为p

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

{ cout<<"请输入第"<<i+1<<"个学生的成绩:";

cin>>p[i]; //向已分配的空间输入数据

s=s+p[i];

}

cout<<"学生的平均成绩为:"<<s/n<<endl;

delete []p; //释放动态分配的存储空间

return 0; }

**课后习题**

P201.8 用指针表示字符串时复制、连接

#include<iostream>

#include<cstring>

using namespace std;

int main()

{ char p1[20]="abcd",\*p2="ABCD";

char str[50]="xyz";

strcpy(str+2,strcat(p1+2,p2+1)); //加几代表从第几个开始操作

cout<<str<<endl;

return 0;

}

答案：xycdBCD

**用指针求字符串长度**

#include<iostream>

using namespace std;

int main()

{ char \*s,\*s1="Here";

s=s1;

while(\*s1) s1++;

cout<<(s1-s)<<endl;

return 0;

}

编程1 用指针将n个数按逆序排列

#include<iostream>

using namespace std;

void inverse(int \*p,int n)

{ int \*q=p+n-1;

int t;

while(p<q)

{ t=\*p;

\*p=\*q;

\*q=t;

p++; q--;

}

}

int main()

{ int \*p,n;

cout<<"请输入一串整数的个数:";

cin>>n;

**p=new int[n];**

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

{ cin>>p[i];}

inverse(p,n);

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

{ cout<<p[i]<<'\t';}

return 0;

}

编程2 编程输入一行文字找其中大写、小写、空格、数字、其他字符个数

#include<iostream>

#include<cstring>

using namespace std;

int main()

{ int n,a=0,b=0,c=0,d=0,e;

char tjt[200];

cout<<"请输入一串文字:";

cin.getline(tjt,200);

**n=strlen(tjt);**

**char \*p;**

**p=&tjt[0];**

while(\*p!='\0')

{

if(\*p>='A'&&\*p<='Z')

{

a++;

}

if(\*p>='a'&&\*p<='z')

{

b++;

}

if(\*p==32)

{

c++;

}

if(\*p>='0'&&\*p<='9')

{

d++;

}

**p++;**

}

e=n-a-b-c-d;

cout<<"大写字母有:"<<a<<"个"<<'\n'<<"小写字母有:"<<b<<"个"<<'\n'<<"空格有:"<<c<<"个"<<'\n'<<"数字字符有:"<<d<<"个"<<'\n'<<"其他字符有:"<<e<<"个";

}

编程3 **指针数组：**编写一个从n个字符串中找最长字符串的函数，并返回最长串的首地址

#include<iostream>

#include<cstring>

using namespace std;

**char \*longstr(char \*z[], int n)**

**{**

**char \*pmaxstr;**

**int len,maxlen;**

**pmaxstr=z[0];**

**maxlen=strlen(z[0]);**

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

**{**

**len=strlen(z[i]);**

**if(maxlen<len)**

**{**

**maxlen=len;**

**pmaxstr=z[i];**

**}**

**}**

**return pmaxstr;**

**}**

int main()

{

char \*pstr[]={"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"};

char \*pmax;

int n;

n=sizeof(pstr)/sizeof(pstr[0]);

pmax=longstr(pstr,n);

cout<<"最长的字符串是: "<<pmax<<endl;

return 0;

}

编程4 函数版：将一个字符串插入到另一个字符串的指定位置

#include<iostream>

#include<cstring>

using namespace std;

**void insert(char str[], char substr[], int n) //将substr插入到str的第n个字符后**

**{**

**char str1[100];**

**int len;**

**len=strlen(str);**

**if(n>=len-1) //想插入的位置在字符串最后一个字母或在其之后**

**{**

**strcat(str,substr); //直接连接**

**return ;**

**}**

**strcpy(str1,str+n); //先将原字符串第n个字符之后的内容保存到新的储存数组中**

**strcpy(str+n,substr); //子串内容覆盖掉原字符串第n个字符之后的内容**

**strcat(str,str1); //将储存数组中的内容拼接到原字符串之后**

**}**

int main()

{

int n;

char str[100],substr[100];

cout<<"请输入第一个字符串: ";

cin.getline(str,100);

cout<<"请输入第二个字符串: ";

cin.getline(substr,100);

cout<<"将字符串"<<substr<<"插入到字符串"<<str<<"的第几个字符后? ";

cin>>n;

insert(str,substr,n);

cout<<"插入后的字符串为: "<<str<<endl;

return 0;

}

**第八章：结构体**

例题8.2 **（结构体数组初始化）**

对班级中的学生姓名按字典顺序排序，并输出排序后的学生学号、姓名、成绩等

#include<iostream >

#include<cstring>

using namespace std;

struct student

{ int id;

char name[20];

char sex;

};

**student num[]={{1011, "Zhang",'F'},{1012, "Liu", 'M'},{1013, "Wang",'M'}};//初始化数组元素**

int main()

{  **int n=sizeof(num)/sizeof(student); //计算数组的元素个数**

student temp;

int i,j;

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

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

if(strcmp(num[j].name ,num[j+1].name )>0) //用字符串处理函数比较name大小

{ temp=num[j]; //数组元素整体互换

num[j]=num[j+1];

num[j+1]=temp;

}

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

{cout<<num[i].id<<'\t'<<num[i].name <<'\t'<<num[i].sex<<endl; }

return 0;

}

例题8.3 **（结构体变量的指针）**

通过结构体类型的指针来引用结构体变量的成员

#include<iostream >

#include<cstring>

using namespace std;

struct student

{ int id;

char name[20];

float score[4];

};

int main()

{ **student stu, \*p;**

**p=&stu; //为结构体指针变量赋值**

**stu.id=1011; //用变量名.成员名的形式引用成员**

strcpy(stu.name,"Zhang");

stu.score[0]=90;

stu.score[1]=95;

stu.score[2]=80;

stu.score[3]=88;

**cout<<p->name<<"的信息如下:\n"; //用指针名->成员名的形式引用成员**

cout<<p->id<<'\t'<<p->name<<'\t';

for(int i=0;i<4;i++)

cout<<p->score[i]<<'\t';

cout<<endl;

return 0;

}

例题8.8 枚举法分析输出结果

#include<iostream>

using namespace std;

int main( )

{ enum team{ qiaut, cubs=4, pick, dodger=qiaut-2};

cout<<qiaut<<'\t'<<cubs<<'\t';

cout<<pick<<'\t'<<dodger<<endl;

return 0;

}

结果：0 4 5 -2

**课后习题**

编程1 用结构体求两点距离

#include<iostream>

#include<cmath>

using namespace std;

struct point

{ double a; //横坐标

double b; //纵坐标

};

int main()

{ **point p1,p2; //定义结构体变量**

cout<<"输入一个横坐标";

cin>> p1.a;

cout<<"输入一个纵坐标";

cin>>p1.b;

cout<<"输入一个横坐标";

cin>>p2.a;

cout<<"输入一个纵坐标";

cin>>p2.b;

cout<<"距离为："<<sqrt((p1.a-p2.a)\*(p1.a-p2.a)+(p1.b-p2.b)\*(p1.b-p2.b));

return 0;

}

**第十章：构造函数和析构函数**

例10.1 **构造函数对数据成员赋值：**用构造函数对对象初始化（9.3改写）

#include<iostream>

#include<cmath>

using namespace std;

class Tri

{ private:

double a,b,c;

public:

Tri(double x,double y,double z) //A构造函数，对三角形三边进行初始化

{a=x; b=y; c=z;} //用形参x、y、z的值初始化三角形三边

double Peri(void)

{return a+b+c;}

double Area(void)

{ double t=(a+b+c)/2;

double s;

s=sqrt(t\*(t-a)\*(t-b)\*(t-c));

return s;

}

};

int main()

{ Tri tria(3,4,5); //B创建三角形对象，调用构造函数对tria三角形的三边初始化

Tri trib(5,5,5); //C调用构造函数对trib三角形的三边初始化

cout<<"tria的周长为:"<<tria.Peri()<<'\t'<<"面积为:"<<tria.Area()<<endl;

cout<<"trib的周长为:"<<trib.Peri()<<'\t'<<"面积为:"<<trib.Area()<<endl;

return 0;

}

例10.2 **构造函数重载**：系统选择根据形式调用哪个构造函数

#include<iostream>

#include<cmath>

using namespace std;

class Tri

{ private:

double a,b,c;

public:

Tri(double x,double y,double z) //A三个参数的构造函数

{a=x; b=y; c=z;}

**Tri(double x) //B一个参数的构造函数**

**{a=b=c=x;}**

double Peri(void)

{return a+b+c;}

};

int main()

{ Tri tria(3,4,5); //C 调用A行构造函数

Tri trib(6); //D 调用B行构造函数

cout<<"tria的周长为:"<<tria.Peri()<<endl;

cout<<"trib的周长为:"<<trib.Peri()<<endl;

return 0;

}

例10.2+ **默认参数的构造函数**

#include<iostream>

#include<cmath>

using namespace std;

class Tri

{ private:

double a,b,c;

public:

**Tri(double x=5,double y=5,double z=5) //参数为默认值的构造函数**

{a=x; b=y; c=z;}

double Peri(void)

{return a+b+c;}

};

int main()

{

Tri tria; //相当于Tri tria(5,5,5)

Tri trib(3); //相当于Tri tria(3,5,5)

Tri tric(3,4); //相当于Tri tria(3,4,5)

Tri trid(3,4,4); //相当于Tri tria(3,4,4)

cout<<"tria的周长为:"<<tria.Peri()<<endl;

cout<<"trib的周长为:"<<trib.Peri()<<endl;

cout<<"tric的周长为:"<<tric.Peri()<<endl;

cout<<"trid的周长为:"<<trid.Peri()<<endl;

return 0;

}

例10.4 分析析构函数的执行顺序

#include<iostream>

using namespace std;

class Tri

{ private:

double a,b,c;

public:

Tri(double x,double y,double z) //A三个参数的构造函数

{ a=x; b=y; c=z;

cout<<"边长是"<<a<<'\t'<<b<<'\t'<<c<<"的三角形调用构造函数"<<endl;

}

~Tri() //析构函数

{cout<<"边长是"<<a<<'\t'<<b<<'\t'<<c<<"的三角形调用析构函数"<<endl;}

};

int main()

{ Tri tria(3,4,5);

Tri trib(6,6,6);

cout<<"main函数"<<endl;

return 0;

}

结果：边长是3 4 5的三角形调用构造函数

边长是6 6 6的三角形调用构造函数

main函数

边长是6 6 6的三角形调用析构函数

边长是3 4 5的三角形调用析构函数

例10.5 设计一个数组类Array，对一维数组进行排序（选择）

#include<iostream>

using namespace std;

class Array

{ int \*p; //数组指针

int n; //数组个数

public:

Array(int a[],int m); //构造函数

void Sort(void); //排序

void Show(void); //显示数组内容

~Array(); //析构函数

} ;

**Array::Array(int a[],int m) //构造函数**

**{ n=m; //用m初始化元素个数n**

**p=new int[n]; //为指针p分配动态存储空间**

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

**\*(p+i)=a[i]; //将数组元素的值赋给数组指针存储**

**}**

void Array::Sort() //选择法由小到大排序

{ int i,j,k,t;

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

{ k=i;

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

{ if(p[k]>=p[j])

k=j;

}

t=p[i];

p[i]=p[k];

p[k]=t;

}

}

void Array::Show() //输出数组元素

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

cout<<p[i]<<'\t';

cout<<endl;

}

**Array::~Array() //析构函数，释放new开辟的存储空间**

**{ if(p) delete []p;}**

int main()

{ int a[]={4,6,2,7,1,8};

**Array arr(a,sizeof(a)/sizeof(a[0])); //用数组a创建数组类对象**

cout<<"原数组为:"<<'\n';

arr.Show();

arr.Sort();

cout<<"排序后的数组为:"<<'\n';

arr.Show();

return 0;

}

**例10.9 复制构造函数**

#include<iostream>

#include<cstring>

using namespace std;

class STR //字符串类

{ char \*p; //字符指针

public:

STR(char \*s);

STR(STR &str);

~STR();

} ;

**STR::STR(char \*s)**

**{ p=new char[strlen(s)+1]; //动态开辟存储空间**

**strcpy(p,s); //对字符串赋值**

**cout<<"调用"<<p<<"的构造函数"<<endl;**

**}**

STR::~STR()

{ if(p)

{ cout<<"调用"<<p<<"的析构函数"<<endl;

delete []p;

}

}

**STR::STR(STR &str) //复制构造函数**

**{ p=new char[strlen(str.p)+1]; //为新建对象动态分配存储空间**

**strcpy(p,str.p); //将原对象的内容复制到新建立的对象**

**cout<<"复制构造"<<p<<endl;**

**}**

int main()

{ STR str1("CHINA");

STR str2(str1);

return 0;

}

结果：调用CHINA的构造函数

复制构造CHINA

调用CHINA的析构函数

调用CHINA的析构函数

面向对象的程序设计实例

**例10.12 对一个一维数组进行排序后实现数值查找**

#include<iostream>

using namespace std;

class Array

{ int a[100]; //待处理数组

int m; //m为成员数组a中元素的个数

public:

Array(int t[],int n); //构造函数，用t初始化a，n初始化m

void sort(); //a中元素从小到大排序

int find(int x); //a中查找数值x

void print(); //输出a

} ;

Array::Array(int t[],int n)

{ m=n;

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

a[i]=t[i];

}

void Array::sort()

{ int i=0,j=0,min,t;

for(i=0;i<m-1;i++)

{ min=i;

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

{ if(a[min]>=a[j])

min=j;

}

t=a[i];

a[i]=a[min];

a[min]=t;

}

}

int Array::find(int x)

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

{ if(x==a[i])

return i;

}

return -1;

}

void Array::print()

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

cout<<a[i]<<'\t';

}

int main()

{ int a[]={2,14,6,18,10,7,9,2,2,35},x;

Array arr(a,sizeof(a)/sizeof(a[0]));

cout<<"排序后的数组为:";

arr.sort();

arr.print();

cout<<endl;

cout<<"请输入要查找的数:";

cin>>x;

x=arr.find(x);

if(x!=-1)

cout<<"查找成功，匹配元素的下标为:"<<x<<endl ;

else cout<<"查找不成功";

return 0;

}

编程2 求100内的无暇素数（正序反序都是素数）

#include<iostream>

using namespace std;

class NUM

{ private:

int a[20]; //整型数组，用来存储100以内的无暇素数

int count; //记录找到的无暇素数的个数

public:

NUM(); //构造函数，将count初始化为0

int reverse(int n); //求取并返回n的逆序数

int isPrime(int n); //判断n是否为素数，若是则返回1，否则返回0

void fun(); //求100以内的所有无暇素数并存储在数组a中

void print(); //输出100以内的无暇素数个数和大小

} ;

NUM::NUM()

{

count=0;

}

**int NUM::reverse(int n) //求整数的反序数**

**{ int m=0;**

**while(n>0)**

**{ m=m\*10+n%10;**

**n=n/10;**

**}**

**return m;**

**}**

int NUM::isPrime(int n)

{ int prime;

prime=1;

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

{ if(n%i==0)

prime=0;

}

return prime;

}

void NUM::fun()

{

int i=0;

for(int j=10;j<100;j++)

{ if(isPrime(j)==1&&isPrime(reverse(j))==1)

{ a[i]=j;

count=count+1;

i++;

}

}

}

void NUM::print()

{ cout<<"count="<<count<<endl;

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

cout<<a[i]<<'\t';

}

int main()

{ NUM num;

NUM();

num.fun();

num.print();

return 0;

}

编程3 将一维数组中的元素按元素各位数字之和从小到大排序

#include<iostream>

using namespace std;

class ARRAY

{ private:

int a[100]; //待排序的数组

int n; //数组中元素的个数

public:

ARRAY(int t[],int m);

int sum(int x); //求整数x的各位数字之和

void fun(); //按要求对数组a的元素排序

void print(); //输出数组a中的所有元素

};

ARRAY::ARRAY(int t[],int m)

{ n=m;

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

a[i]=t[i];

}

int ARRAY::sum(int x)

{  **int i=0,z=0; //i用于表示位数,z用于求各位数字之和**

**int k=x;**

**int t=x;**

**do**

**{ t=k%10;**

**i=k/10;**

**k=k/10;**

**z=z+t;**

**}while(i>0);**

**return z;**

}

void ARRAY::fun()

{ int t;

int max;

for(int i=0;i<n-1;i++)

{ int min=i;

for(int j=i+1;j<n;j++)

{ if(sum(a[min])>=sum(a[j]))

min=j;

}

t=a[i];

a[i]=a[min];

a[min]=t;

}

}

void ARRAY::print()

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

cout<<a[i]<<'\t';

}

int main()

{ int b[]={297,735,624,158,312,900};

ARRAY arr(b,sizeof(b)/sizeof(b[0]));

cout<<"排序前的数组为:";

arr.print();

cout<<endl;

arr.fun();

cout<<"排序后的数组为:";

arr.print();

return 0;

}

例10.15 将字符串表示的16进制数转换为10进制数

#include<iostream>

#include<cstring>

using namespace std;

class Num //将16进制转换成10进制类

{ char \*p; //字符串，存放16进制数

int n; //存放转换后的10进制数

public:

Num(char \*str); //构造函数，用str初始化p

void fun(); //转换

void print(); //输出16进制和对应的10进制数

~Num();

} ;

Num::Num(char \*str)

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

strcpy(p,str);

}

void Num::fun()

{  **n=0;**

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

**{ if(p[i]>='0'&&p[i]<='9') //16进制字符为数字字符**

**n=n\*16+p[i]-'0';**

**else if(p[i]>='a'&&p[i]<='f') //16进制字符为小写字符**

**n=n\*16+p[i]-'a'+10;**

**else n=n\*16+p[i]-'A'+10; //16进制字符为大写字符**

**}**

}

void Num::print()

{ cout<<"16进制数为:0X"<<p<<endl;

cout<<"10进制数为:"<<n<<endl;

}

Num::~Num()

{if(p) delete []p;}

int main()

{ char str[100];

cout<<"请输入一个16进制数:";

cin.getline(str,100);

Num num(str); //初始化转换类的对象

num.fun(); //转换

num.print(); //输出 return 0; }