**实验1 函数重载和函数模板**

**一、实验目的：**

1. 理解函数重载和函数模板的概念和意义
2. 会采用函数重载和函数模板进行程序设计
3. 掌握不同函数模板的用法

**二、实验内容和要求**

1、

#include<iostream>

using namespace std;

class in

{

    public:

    /\*定义int相加的函数\*/

    void input(int a,int b)

    {

        int c=a+b;

        cout<<"int型相加:"<< c << endl;

    }

     /\*重载float相加的函数\*/

    void input(float a,float b)

    {

        float c=a+b;

        cout<<"float型相加:"<< c << endl;

    }

     /\*重double相加的函数\*/

    void input(double a,double b)

    {

        double c=a+b;

        cout<<"double型相加:"<< c << endl;

    }

     /\*重载一维数组相加的函数\*/

    void input(int a[],int b[],int n)

    {

        int i;

        int c[n];

        for(i=0;i<n;i++)

        {

            c[i]=a[i]+b[i];

        }

        cout<<"一维数组相加:";

        for(i=0;i<n;i++)

        {

            cout<<c[i]<<" ";

        }

        cout<<endl;

    }

     /\*重载二维数组相加的函数\*/

    void input(int \*a[],int \*b[],int n,int m)

    {

        int i,j;

        int c[n][m];

        cout<<"二维数组相加:\n";

        for(i=0;i<n;i++)

        {

            for(j=0;j<m;j++)

            {

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

                cout<<c[i][j]<<" ";

            }

            cout<<"\n";

        }

        cout<<endl;

    }

};

main()

{

    in i;

    int a=1,b=2;

    float c=2.2,d=3.3;

    double e=4.14,f=5.25;

    int p[4]={1,2,3,4};

    int q[4]={2,3,4,5};

    int x[3][3]={1,2,3,4,5,6,7,8,9};

    int y[3][3]={2,3,4,5,6,7,8,9,10};

    int \*u[3],\*v[3];

    for(int t=0;t<3;t++)

    {

        u[t]=&x[t][0];

        v[t]=&y[t][0];

    }

    i.input(a,b);

    i.input(c,d);

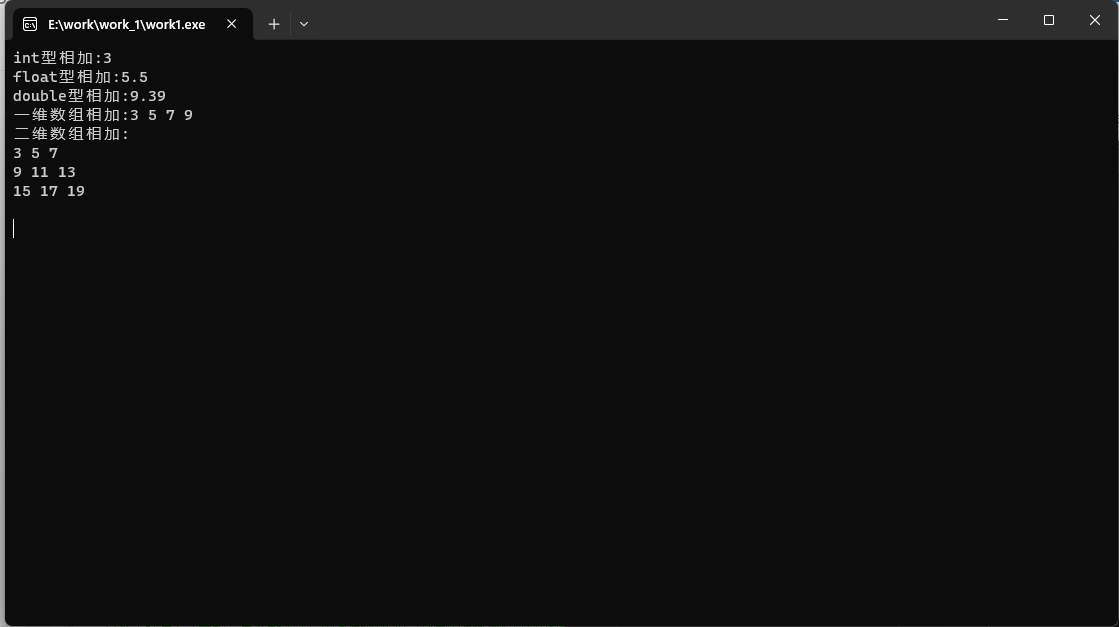
    i.input(e,f);

    i.input(p,q,4);

    i.input(u,v,3,3);

    return 0;

}



2、

#include<iostream>

using namespace std;

template<typename T,typename T2>//区分两种不同类型比较

T max(T a,T2 b)

{

    T c=(a > b ? a : b);

    return c;

}

main()

{

    int a=1;

    float b=2.0;

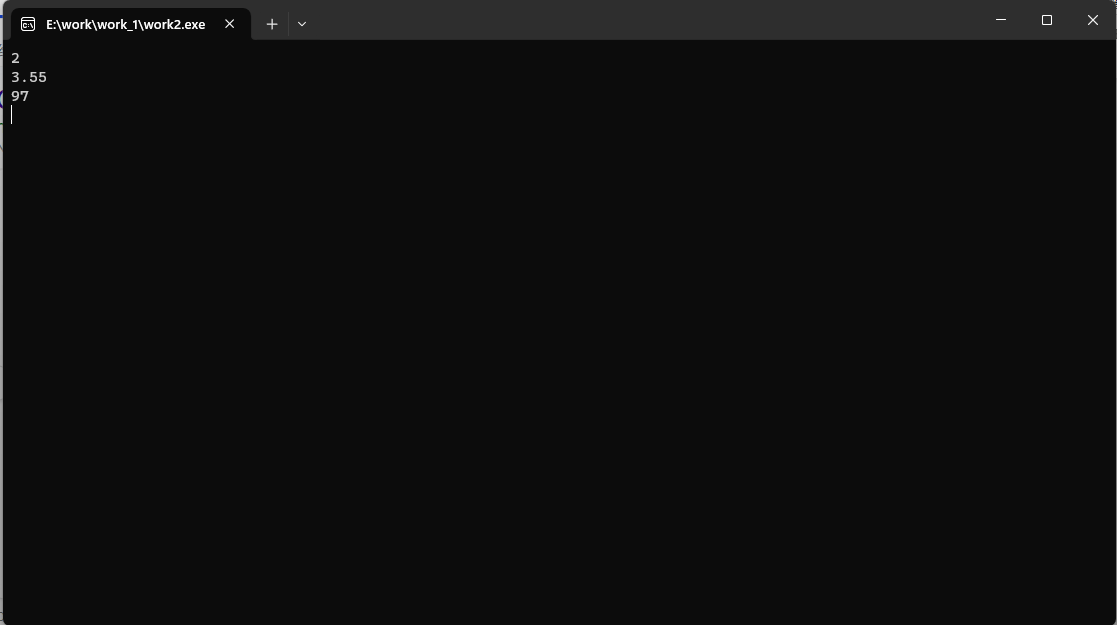
    double c=3.55;

    char d='a';

    cout<<max(a,b)<<'\n'<<max(b,c)<<'\n'<<max(a,d)<<endl;

    return 0;

}



3、

#include<iostream>

using namespace std;

template <typename T>

void maxMin(T \*A[],int m,int n)

{

    int i,j;

    T max,min;

    max=A[0][0];

    min=A[0][0];

    for(i=0;i<m;i++)

    {

        for(j=0;j<n;j++)

        {

            if(A[i][j]>max)

            {

                max=A[i][j];

            }

            if(A[i][j]<min)

            {

                min=A[i][j];

            }

        }

    }

    cout<<"最大值为"<<max<<"\n"<<"最小值为"<<min<<endl;

}

main()

{

    int a[2][3]={2,4,5,1,6,7};

    int \*a0[2];

    double b[2][3]={1.1,2.2,3.4,5.2,4.6,9.7};

    double \*b0[2];

    for(int i=0;i<2;i++)

    {

        a0[i]=&a[i][0];

        b0[i]=&b[i][0];

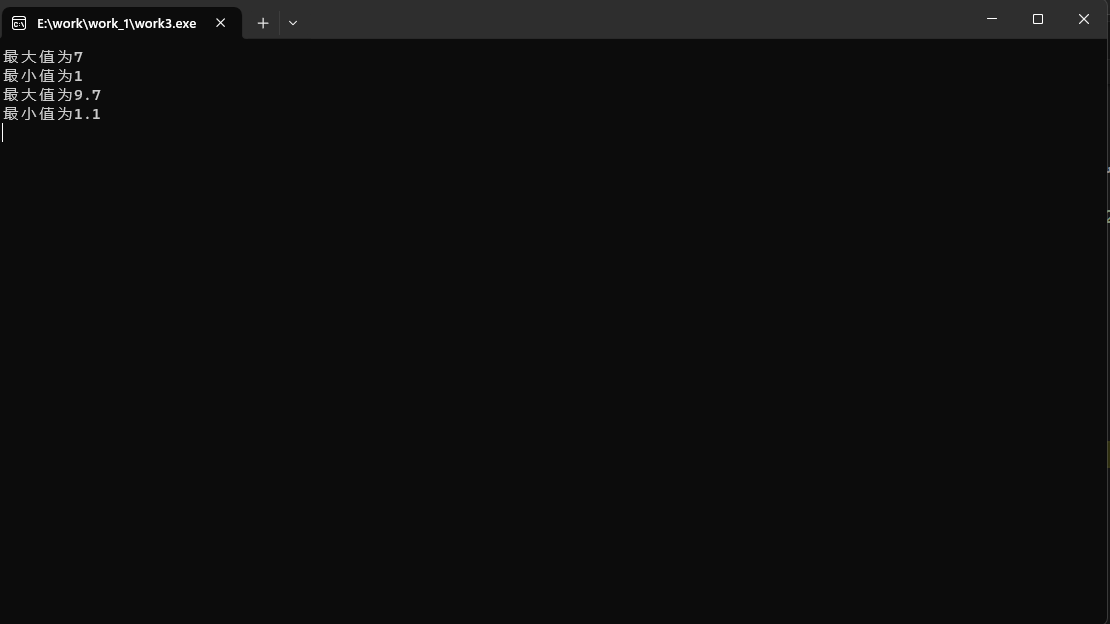
    }

    maxMin(a0,2,3);

    maxMin(b0,2,3);

    return 0;

}



三、问题回答

1）在函数参数不同且需要同名函数（比如构造函数），使用函数重载；在需要重复使用一个函数过程但函数参数类型不同时使用函数模板。

2）函数重载必须返回值相同；成员函数名称和参数不能完全相同； 函数模板的调用实际上是产生新的函数体。