

```

#pragma once
#include<iostream>
using namespace std;
enum signType{PLUS,MINUS};
typedef unsigned long ul;
typedef unsigned int ui;
class Currency
{
public:
    Currency (signType thesign,ul thedollars=0,ui cents=0);
    Currency(){};
    ~Currency(){}
    //注意,此时的函数都不包含具体变量名,只有数据类型
    void setValue(signType,ul,ui );
    void setValue(double);

    //返回符号,美元和美分
    signType getSign() const {return sign;}
    ul getDollars()const {return dollars;}
    ui getCents()const {return cents;}

    //两个加函数
    Currency add(const Currency&) const ;
    Currency& increment(const Currency&);

    //输出
    void output()const ;

    //(1)输入
    void input();
    //(2)减法
    double subtract(double x)const ;
    //(3)%x
    Currency percent(double x)const;
    //(4)乘法
    Currency multiply(double x)const;
    //(5)除法
    Currency divide(double x)const ;
private:
    signType sign;
    ul dollars=0;
    ui cents=0;
};

```

```

//(1)输入
void Currency::input()
{
    double value;
    cin>>value;
    //处理符号
    if(value<0)
    {
        sign=MINUS;
        value=-value;
    }
    else sign=PLUS;

    dollars=ul(value);
    cents=ui((value-dollars+0.001)*100);
}

//(2)减法
double Currency::subtract(double x)const
{
    double temp=(dollars*100+cents);
    if(sign==MINUS)temp=-temp;
    return temp/100-x;
}

//(3)%x
Currency Currency::percent(double x)const
{
    Currency result;
    double temp=(dollars*100+cents)*x/100.0;
    result.dollars=temp/100;
    //处理符号
    if(temp>0)
        result.sign=PLUS;
    else result.sign=MINUS;

    result.cents=temp-result.dollars*100;
    return result;
}

//(4)乘法
Currency Currency::multiply(double x)const
{
    Currency result;
    double temp=(dollars*100+cents)/100.0;

```

```

        if(sign==MINUS)temp=-temp;
        double res=temp*x;

        result.dollars=(ul)abs(res);

        //处理符号
        if(res>0)
            result.sign=PLUS;
        else result.sign=MINUS;

        result.cents=(ui)((abs(res)-result.dollars+0.001)*100);

        return result;
    }

// (5) 除法, 相比乘法只将*变为/
Currency Currency::divide(double x) const
{
    Currency result;
    double temp=(dollars*100+cents)/100.0;
    if(sign==MINUS)temp=-temp;
    double res=temp/x;

    result.dollars=(ul)abs(res);

    //处理符号
    if(res>0)
        result.sign=PLUS;
    else result.sign=MINUS;

    result.cents=(ui)((abs(res)-result.dollars+0.001)*100);

    return result;
}

int main()
{
    //测试数据
    Currency g;
    g.input();
    g.output();
    cout<<g.subtract(1.2)<<endl;
    g.percent(1).output();
    g.multiply(-2).output();
}

```

```

    g.divide(2).output();

    //输入:
    //4.6
    // 输出:
    // $4.60
    // 3.4
    // $0.04
    // -$9.20
    // $2.30
    return 0;
}

```

P29 23

```

#include<iostream>
using namespace std;

// (1)
// gcd(20,30)=gcd(30,20)=gcd(20,10)=gcd(10,0)=10
// gcd(112,42)=gcd(42,28)=gcd(28,14)=gcd(14,0)=14

// (2)
// 基础部分:y=0,g(x,y)=x
// 递归部分:y!=0 gcd(x,y)=gcd(y,x mod y)

// 证明
// 如果 d 是 a 和 b 的公约数, 那么 d 也是 r1 和 b 的公约数,
// 因为 a 和 b 之间的关系可以表示为  $a = b * q1 + r1$ , 其中 q1 是整数。
// 因此, d 也必须是 r2 和 r1 的公约数, 以此类推, 直到余数为 0。所以, d 是 a 和 b 的
// 最大公约数。

int gcd(int x,int y)
{
    if(y==0)return x;
    return (gcd(y,x%y));
}

int main()
{
    cout<<gcd(0,10)<<endl;
    cout<<gcd(20,30)<<endl;
    cout<<gcd(112,42)<<endl;
    //输出结果 10 10 14
}

```

```
#include<iostream>

using namespace std;
template <class T>
bool is_in(T x,T array[],int pos,int n)
{
    if(pos>=n)return false;
    if(array[pos]==x&&pos<n)
        return true;
    else
        return is_in(x,array,pos+1,n);
}

int main()
{
    int array[]={1,4,5,3,4,7,6};
    int x;cin>>x;
    int n=sizeof(array)/sizeof(array[0]);
    if(is_in<int>(x,array,0,n))
        cout<<"Yes"<<endl;
    else cout<<"No"<<endl;

    return 0;
    //测试: 6 YES 2 NO
}
```