

## چند مضربی

به شما سه کلاس A , B , C داده میشود . هر سه این کلاس ها تابع func مخصوص به خود را دارند.

در کلاس A : این تابع مقداری که به آن پاس داده میشود را دوباربر میکند.

```
class A
{
    public:
        A(){
            callA = 0;
        }
    private:
        int callA;
        void inc(){
            callA++;
        }

    protected:
        void func(int & a)
        {
            a = a * 2;
            inc();
        }
    public:
        int getA(){
            return callA;
        }
};
```

در کلاس B : این تابع مقداری که به آن پاس داده میشود را سه برابر میکند.

```
class B
{
    public:
        B(){
            callB = 0;
        }
};
```

```
    }  
private:  
    int callB;  
    void inc(){  
        callB++;  
    }  
protected:  
    void func(int & a)  
    {  
        a = a * 3;  
        inc();  
    }  
public:  
    int getB(){  
        return callB;  
    }  
};
```

در کلاس C : این تابع مقداری که به آن پاس داده میشود را پنج برابر میکند.

```
class C  
{  
    public:  
        C(){  
            callC = 0;  
        }  
    private:  
        int callC;  
        void inc(){  
            callC++;  
        }  
    protected:  
        void func(int & a)  
        {  
            a = a * 5;  
            inc();  
        }  
    public:  
        int getC(){  
            return callC;  
        }  
};
```

```
    }
};
```

به شما کلاس D به فرم زیر داده میشود :

```
class D
{
    int val;
public:
    //Initially val is 1
    D()
    {
        val = 1;
    }

    //Implement this function
    void update_val(int new_val)
    {

    }

    //For Checking Purpose
    void check(int); //Do not delete this line.
};
```

حال شما باید تابع update\_val را بگونه ای بنویسید که مقدار val موجود در کلاس D را به new\_val تغییر دهد . این کار باید تنها با صدا زدن توابع  $\$func\$$  موجود در کلاس های A , B , C انجام دهید. تضمین میشود که new\_val تنها مضربی از 2 ، 3 ، 5 باشد .

## ورودی :

حاوی تنها یک خط است که مقدار new\_val را از کاربر میگیرد

## خروجی :

به صورت اتوماتیک توسط کد زیر خروجی داده خواهد شد که فرمت نمونه آن در مثال ها موجود میباشد .

مثال :

ورودی :

30

خروجی :

```
Value = 30
A's func called 1 times
B's func called 1 times
C's func called 1 times
```

توضیحات :

در ابتدا مقدار val یک میباشد . سپس تابع func موجود در کلاس A اجرا میشود و مقدار val دوبار برابر میشود. سپس تابع func موجود در کلاس B اجرا میشود و مقدار val سه برابر میشود. سپس تابع func موجود در C اجرا میشود و این بار 5 برابر میشود که میشود 30 !

کد خروجی شما باید به فرم زیر باشد :

```
#include<iostream>

using namespace std;

class A
{
public:
    A(){
        callA = 0;
    }
private:
    int callA;
    void inc(){
        callA++;
    }
};
```

```
    }

protected:
    void func(int & a)
    {
        a = a * 2;
        inc();
    }
public:
    int getA(){
        return callA;
    }
};

class B
{
public:
    B(){
        callB = 0;
    }
private:
    int callB;
    void inc(){
        callB++;
    }
protected:
    void func(int & a)
    {
        a = a * 3;
        inc();
    }
public:
    int getB(){
        return callB;
    }
};

class C
{
public:
    C(){
```

```

        callC = 0;
    }
private:
    int callC;
    void inc(){
        callC++;
    }
protected:
    void func(int & a)
    {
        a = a * 5;
        inc();
    }
public:
    int getC(){
        return callC;
    }
};

/*****/

class D
{
    int val;
public:
    //Initially val is 1
    D()
    {
        val = 1;
    }

    //Implement this function
    void update_val(int new_val)
    {

    }

    //For Checking Purpose
    void check(int); //Do not delete this line.
};

```

```

/*****/

void D::check(int new_val)
{
    update_val(new_val);
    cout << "Value = " << val << endl << "A's func called " << getA() <
}

int main()
{
    D d;
    int new_val;
    cin >> new_val;
    d.check(new_val);

}
```