

Overload Operators

You are given a class - *Complex*.

```
class Complex
{
public:
    int a,b;
};
```

Operators are overloaded by means of operator functions, which are regular functions with special names. Their name begins with the operator keyword followed by the operator sign that is overloaded. The syntax is:

```
type operator sign (parameters) { /*... body ...*/ }
```

You need to overload operators `+` and `<<` for the *Complex* class.

The operator `+` should add complex numbers according to the rules of complex addition:

$$(a+ib)+(c+id) = (a+c) + i(b+d)$$

Overload the stream insertion operator `<<` to add "*a + ib*" to the stream:

```
cout<<c<<endl;
```

The above statement should print "*a + ib*" followed by a newline where *a* = *c.a* and *b* = *c.b*.

Input Format

The overloaded operator `+` should receive two complex numbers (*a + ib* and *c + id*) as parameters. It must return a single complex number.

The overloaded operator `<<` should add "*a + ib*" to the stream where *a* is the real part and *b* is the imaginary part of the complex number which is then passed as a parameter to the overloaded operator.

Output Format

As per the problem statement, for the output, print "*a + ib*" followed by a newline where *a* = *c.a* and *b* = *c.b*.

Sample Input

```
3+i4
5+i6
```

Sample Output

$8+i10$

Explanation

Given output after performing required operations (overloading + operator) is $8+i10$.