Task 1:

1. Construct a class *CSetInt* which represents a set of integer numbers (the numbers may not repeat). Implement the following functions:

   - *void Insert(int n);* – to include a new element in the set. If the element *n* is already included in the set, nothing has to be done. Initially, it is unknown how many elements will be included in the set;
   - *bool Contains(int n);* – to check whether the set contains a given element *n*;
   - *void Remove(int n);* - to remove an element *n* from the set. If the set does not include element *n*, nothing has to be done;
   - *void Empty();* - to delete all the elements from the set;
   - *void Display();* - to display all the elements of the set on the monitor screen;
   - *unsigned Size();* - to calculate the number of elements included in the set.

2. Add the following functions to the class *CSetInt*:

   - *CSetInt Union(const CSetInt& add);* - to create a new set which is a union of **this** and *add*. The resulting set should not contain repeating elements.
   - *CSetInt Difference(const CSetInt& dif);* - to create a new set which is a difference of **this** and *dif*.
   - *CSetInt Intersection(const CSetInt& inter);* - to create a new set which is an intersection of **this** and *inter*. The resulting set should not contain repeating elements.

3. Decide which of the listed above member functions are to be declared **const**.

4. Test the developed class with the aid of the following *main* function:

```
int main(int argc, char* argv[])
{
	using namespace std;

	CSetInt c1;
	c1.Insert(4); c1.Insert(7); c1.Insert(6); c1.Insert(5);

	CSetInt c2 = c1;
	c2.Insert(3); c2.Insert(2); c2.Insert(4); c2.Insert(7);
	c2.Remove(3);  c2.Remove(5); c2.Remove(6);

	c1.Display();c2.Display();

	cout << "Number of elements in c1: " << c1.Size() << endl;
	cout << "Number of elements in c2: " << c2.Size() << endl;

	cout << "c1 includes 6?: " << (c1.Contains(6) ? "yes" : "no") << endl;
	cout << "c2 includes 6?: " << (c2.Contains(6) ? "yes" : "no") << endl;

	cout << "Union:" << endl;
	c1.Display(); c2.Display();
```

```
        CSetInt c3 = c1.Union(c2);        c3.Display();

        c2.Insert(3);
        cout << "Intersection:" << endl;
        c1.Intersection(c2).Display();

        cout << "Difference:" << endl;
        c2.Difference(c1).Display();

        c1.Empty();  c1.Display();

        return 0;
    }
```

The results should be the following (the order of elements is irrelevant):

```
Set elements:  4 7 6 5

Set elements:  4 7 2

Number of elements in c1: 4
Number of elements in c2: 3
c1 includes 6?: yes
c2 includes 6?: no

Union:
Set elements:  4 7 6 5
Set elements:  4 7 2
Set elements:  4 7 6 5 2

Intersection:
Set elements:  4 7

Difference:
Set elements:  2 3

Set elements:
```

Task 2:

Answer whether the definition of the following *inc* function is correct:

```
int& inc (int v)
{
      v++;
      return v;
}
```