set_difference 和 includes

set_difference()算法计算两个集合[start1, end1)和[start2, end2)的差集, 并将差集存放到result.

两个集合以序列的形式给出, 且必须先按升序排好位置.

set_difference()是一个指向result序列末尾的迭代器.

如果严格弱排序的比较函数对象cmp未指定, set_difference()将使用<操作符比较元素.

范例

```
// set_difference example
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;

int main () {
  int first[] = {5,10,15,20,25};
  int second[] = {50,40,30,20,10};
  vector<int> v(10);                           // 0  0  0  0  0  0  0  0  0  0
  vector<int>::iterator it;

  sort (first,first+5);     //  5 10 15 20 25
  sort (second,second+5);   // 10 20 30 40 50

  it=set_difference (first, first+5, second, second+5, v.begin());
                                               // 5 15 25  0  0  0  0  0  0  0

  cout << "difference has " << int(it - v.begin()) << " elements.\n";

  return 0;
}
```

另外一个例子是：

```
#include <iostream>

#include <algorithm>
#include <vector>
using namespace std;

bool isOld(int val)
{
    return val%2;
}

bool calF(int nn){

    return nn>7;
}


void disp(int nn){
    cout << nn << endl;
}


int main()
{
    vector<int> vec={1,7,3,9,5};
    vector<int> vec2={5,9,3,11};
    bool h = std::all_of(vec.begin(), vec.end(), isOld);
    cout <<"all of vec is old ? " <<h<<endl;
    auto itt = find_if(vec.begin(),vec.end(), calF);
    for(auto uu = itt; uu != vec.end(); ++uu){
        cout << *uu << endl;
    }
    vector<int> gg;
    set_difference(vec.begin(), vec.end(), vec2.begin(), vec2.end(),back_inserter(gg));
    cout <<"set different "<< endl;
    for_each(gg.begin(), gg.end(), disp);

    cout <<"find diff"<<endl;
    cout <<includes(vec.begin(),vec.end(),vec2.begin(),vec2.end())<<endl;

}
```