```
========================================= algorithm.cpp
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>

template < class iter > void
view (iter b, iter e)
{
  iter it;
  for (it = b; it != e; it++) {
    cout << *it << " ";
  }
  cout << "\n";
}

#define BE(v) v.begin(), v.end()

void
action (int a)
{
  cout << a * a << " ";
}

bool
isodd (int a)
{
  if (a % 2)
    return true;
  return false;
}

main ()
{
  vector < int >v;
  v.push_back (3);
  v.push_back (4);
  v.push_back (1);
  view (BE (v));
  sort (BE (v));                  // sort a vector
  view (BE (v));
  vector < int >::iterator it = lower_bound (BE (v), 2);
  v.insert (it, 2);
  view (BE (v));
  reverse (BE (v));               // reverse a vector
  view (BE (v));
  for_each (BE (v), action);      // do something on each element
  cout << "\n";
  char s[] = "badc";              // sort a string
  sort (s, s + strlen (s));
  cout << s << "\n";
  it = remove_if (BE (v), isodd);       // odd elements at end
  view (BE (v));
  v.erase (it, v.end ());         // erase them from the end
  view (BE (v));
}

---- Output:------------------------------
p3 4 1
1 3 4
1 2 3 4
4 3 2 1
```

```
16 9 4 1
abcd
4 2 2 1
4 2

========================================== io.cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>

template < class iter > void
view (iter b, iter e)
{
  iter it;
  for (it = b; it != e; it++) {
    cout << *it << " ";
  }
  cout << "\n";
}

#define BE(v) v.begin(), v.end()

void
twice (int &a)
{
  a = a * 2;
}

main ()
{
  ifstream data ("data.txt");
  vector < int >v;
  copy (istream_iterator < int >(data), // which file
        istream_iterator < int >(),     // until the end
        back_inserter (v)       // put it into v
    );
  for_each (BE (v), twice);
  copy (BE (v), ostream_iterator < int >(cout, "\n"));
}

---- Output:------------------------------
p2
4
6
8
10
12

========================================== io2.cpp
#include <iostream>
#include <fstream>

main ()
{
  int x, y;
  ifstream data ("data.txt");
  while (data >> x) {
    data >> y;
    cout << x << " " << y << "\n";
  }
}
```

```
---- Output:-----------------------------
p1 2
3 4
5 6

========================================== map.cpp
#include <iostream>
#include <string>
#include <map>

void
view (map < string, int >m)
{
  map < string, int >::iterator it;
  for (it = m.begin (); it != m.end (); it++) {
    cout << (*it).first << "->" << (*it).second << " ";
  }
  cout << "\n";
}

main ()
{
  map < string, int >m;
  m["one"] = 1;
  cout << m["one"] << "\n";
  m.insert (make_pair ((string) "five", 5));
  view (m);
}

---- Output:-----------------------------
p1
five->5 one->1

========================================== pair.cpp
#include <iostream>
#include <utility>

main ()
{
  pair < int, double >p, q, r (2, 2.0);
  p.first = 1;
  p.second = 2.0;
  q = make_pair (1, 3.0);
  cout << p.first << "\n";
  cout << (p < q) << "\n";
  cout << (q == r) << "\n";
}

---- Output:-----------------------------
p1
1
0

========================================== set.cpp
#include <iostream>
#include <set>
#include <algorithm>

void
view (set < int >v)
{
  set < int >::iterator it;
```

```
  for (it = v.begin (); it != v.end (); it++) {
    cout << *it << " ";
  }
  cout << "\n";
}

main ()
{
  set < int >s;
  s.insert (7);
  s.insert (5);
  s.insert (6);
  view (s);
  set < int >::iterator it = s.find (7);
  cout << "Value: " << *it << "\n";
  cout << "Count: " << s.count (7) << "\n";
  cout << "Position: " << distance (s.begin (), it) << "\n";
  advance (it, -2);
  cout << "Before previous: " << *it << "\n";
  s.erase (7);
  view (s);
  s.erase (s.begin ());
  view (s);

  cout << "---\n";

  set < int >t, sum;
  t.insert (4);
  set_union (s.begin (), s.end (),
             t.begin (), t.end (), inserter (sum, sum.begin ()));
  view (sum);
}

---- Output:------------------------------
p5 6 7
Value: 7
Count: 1
Position: 2
Before previous: 5
5 6
6
---
4 6

========================================== set2.cpp
#include <iostream>
#include <set>
#include <algorithm>

typedef struct
{
  int x;
  int y;
}
Tpoint;

bool
operator < (Tpoint p, Tpoint pp)
{
  if (p.x < pp.x)
    return true;
  if (p.x > pp.x)
    return false;
```

```
  if (p.y < pp.y)
    return true;
  return false;
}

set < Tpoint > s;

main ()
{
  set < Tpoint > s;
  Tpoint point;
  point.x = 4;
  point.y = 5;
  s.insert (point);
  s.insert (point);
  cout << s.size ();
  cout << (*s.begin ()).x;
  cout << s.count (point);
}

---- Output:------------------------------
p141
========================================== string.cpp
#include <iostream>
#include <string>


main ()
{
  string s;
  s = "first";

  string ss ("second");

  string sss (5, 'a');          // fill with a character
  cout << sss << "\n";

  char olds[] = "old style";    // interaction with character arrays
  ss = olds;
  strcpy (olds, s.c_str ());
  cout << ss << "\n";
  cout << olds << "\n";

  string t (olds + 2, olds + 4);        // init with pointers
  cout << t << "\n";

  string tt (s.begin () + 2, s.begin () + 4);   // init /w iterators
// string tt(s.begin(),s.end());
  cout << tt << "\n";
  cout << tt.size () << "\n";
  cout << tt[1] << "\n";

  cout << (s < ss) << "\n";     // lexicographic comparison
}

---- Output:------------------------------
paaaaa
old style
first
rs
rs
2
s
```

```
1

========================================= stringstream.cpp
// ######################## stringstream.cpp

#include <iostream>
#include <sstream>

main ()
{
  int i = 5;
  stringstream ss;
  ss << "i= " << i;
  cout << ss.str () << "\n";

  stringstream sss ("10");
  sss >> i;
  cout << i * 2 << "\n";

  sss.str ("-10");
  sss >> i;
  cout << i * 2;
}

---- Output:------------------------------
pi= 5
20
-20
========================================= templates.cpp
// ######################## templates.cpp

#include <iostream>
#include <vector>
#include <set>

template < class T > T twice (T a)
{
  return 2 * a;
}

template < class iter > void
view (iter b, iter e)
{
  iter it;
  for (it = b; it != e; it++) {
    cout << *it << "\n";
  }
  cout << "\n";
}

#define BE(v) v.begin(), v.end()

main ()
{
  cout << twice (2) << "\n";
  double a = 3.0;
  cout << twice (a) << "\n";

  cout << "\n";

  set < int >s;
  s.insert (7);
  s.insert (6);
```

```
  view (s.begin (), s.end ());

  vector < int >v (3, 2);
  view (BE (v));
}

---- Output:------------------------------
p4
6

6
7

2
2
2


========================================= vector.cpp
#include <iostream>
#include <vector>

void
view (vector < int >v)
{
  for (int i = 0; i < v.size (); i++) {
    cout << v[i] << " ";
  }
  cout << "\n";
}

void
view2 (vector < int >v)
{
  vector < int >::iterator it;
  for (it = v.begin (); it != v.end (); it++) {
    cout << *it << "\n";
  }
}

void
rview (vector < int >v)
{
  vector < int >::reverse_iterator it;
  for (it = v.rbegin (); it != v.rend (); it++) {
    cout << *it << " ";
  }
  cout << "\n";
}

void
view3 (vector < int >v)
{
  copy (v.begin (), v.end (), ostream_iterator < int >(cout, "\n"));
}

main ()
{
  vector < int >v (3, 2);
  v.push_back (1);
  view (v);
  rview (v);
  cout << "Size " << v.size () << "\n";
```

```
  cout << "Last " << v.back () << " First " << v[0] << "\n";
  v.pop_back ();
  view (v);
  v.clear ();
  v.insert (v.begin (), 2);
  *back_inserter (v) = 1;
  v.insert (v.begin () + 1, 0);
  view (v);
  v.erase (v.begin () + 1);
  view (v);
}

---- Output:------------------------------
p2 2 2 1
1 2 2 2
Size 4
Last 1 First 2
2 2 2
2 0 1
2 1
```