

## Contents

1	section1	1	16
1.1	basic	1	17
1.2	test	1	18
2	Libraries	1	18
2.1	algorithm	1	1
3	Section2	1	19
3.1	thm	1	20

## 1 section1

### 1.1 basic

```

1 // c++ code
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 int main() {
6     // test comment
7     cout << "test string\n";
8 }

```

### 1.2 test

```

1 // map::begin/end
2 #include <iostream>
3 #include <map>
4
5 using namespace std;
6
7 int main ()
8 {
9     map<char,int> mymap;
10
11     mymap['b'] = 100;
12     mymap['a'] = 200;
13     mymap['c'] = 300;
14
15     // show content:
16     for (map<char,int>::iterator it = mymap.begin(); it
17         != mymap.end(); ++it)
18         cout << it->first << " => " << it->second << '\n';
19
20     return 0;
21 }

```

## 2 Libraries

### 2.1 algorithm

```

1 #include <algorithm>
2 using namespace std;
3
4 int main() {
5
6     void sort (RandomAccessIterator first,
7         RandomAccessIterator last);
8     /* Sorts the elements in the range [first,last)
9        into ascending order
10        * In N*lg(N) complexity
11        */
12
13     ForwardIterator lower_bound (ForwardIterator
14         first, ForwardIterator last, const T& val);
15     /* Returns an iterator pointing to the first
16        element in the range [first,last) which is >=
17        val

```

```

13     * In lg(N)+1 complexity
14     * requires sorted elements
15     */
16
17     ForwardIterator upper_bound (ForwardIterator
18         first, ForwardIterator last, const T& val);
19     /* Returns an iterator pointing to the first
20        element in the range [first,last) which is >
21        val
22     * In lg(N)+1 complexity
23     * requires sorted elements
24     */
25
26     pair<ForwardIterator,ForwardIterator> equal_range
27         (ForwardIterator first, ForwardIterator last,
28         const T& val);
29     /* Returns the bounds of the subrange with all
30        the elements == val of the range [first,last)
31        * return type equivalent to pair<
32        lower_bound(), upper_bound>
33     * In 2*lg(N)+1 complexity
34     * requires sorted elements
35     */
36
37     bool next_permutation (BidirectionalIterator
38         first, BidirectionalIterator last);
39     /* Rearranges the elements in the range
40        [first,last) into the next lexicographically
41        greater permutation, then returns
42        * true if could rearrange as a
43        lexicographically greater permutation
44        * false if no greater arrangement than the
45        previous (and sorted in ascending order)
46     * In N/2 complexity
47     */
48
49     bool prev_permutation (BidirectionalIterator
50         first, BidirectionalIterator last);
51     /* Rearranges the elements in the range
52        [first,last) into the previous
53        lexicographically-ordered permutation, then
54        returns
55        * true if could rearrange as a
56        lexicographically smaller permutation
57        * false if arrangement is the largest
58        possible (and sorted in descending order)
59     * In N/2 complexity
60     */
61
62     return 0;
63 }

```

## 3 Section2

### 3.1 thm

- 中文測試

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$