

a)

class Calendar {

int year;

multimap&lt;int, string&gt; events;

int months[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

public:

Calendar &amp; operator=(Calendar c2) {

year = c2.year;

events = c2.events;

return \*this;

}

vector&lt;string&gt; getEvents(int day, int months) {

int dayOfYear = 0;

for (int i = 0; i &lt; month - 1; i++) {

dayOfYear += months[i];

}

dayOfYear += day;

typedef std::multimap&lt;int, string&gt;::iterator mapIt;

pair&lt;mapIt, mapIt&gt; result = events.equal\_range(dayOfYear);

vector&lt;string&gt; dayEvents = {};

for (mapIt it = result.first; it != result.second; it++)

dayEvents.push\_back(it-&gt;second);

return dayEvents;

}

};



б) set <int> v {...};

1) set <int>::reverse\_iterator rit;  
for (rit = v.rbegin(); rit != v.rend(); rit++)  
cout << \*rit << " ";

2) for (auto it = v.begin(); it != v.end(); ++it) {  
if (\*it % 2) {  
cout << \*it << " ";  
}  
}

3) for\_each(v.cbegin(), v.cend(), [](const auto& a) {  
if (a % 2 != 0) cout << a << " "; });

с) #include <iostream>  
#include <mutex>  
#include <thread>  
using namespace std;  
int main() {  
mutex mtx;  
thread thread1([&mtx]() {  
while (true) {  
mtx.lock();  
cout << rand() % 10 << " ";  
mtx.unlock();  
}  
});  
thread thread2([&mtx]() {  
while (true) {  
mtx.lock();  
cout << char('A' + rand() % 26) << " ";  
mtx.unlock();  
}  
});  
thread1.join();  
thread2.join();  
}