ВПЧ-1

Ахпи

а) pubic:

1) Calendar& operator = (const Calendar& calendar){
    year = calendar.year;
    events = calendar.events;
    return *this;
}

2) vector<string> getEvents (int day; string months){
map<string, int> months = { pair<string, int> ("январь", 0), ...
... pair<string, int> ("декабрь", 11)};
if (months.count(months) == 0 || day >= 30 || day < 0){
~~return~~ vector<string> a;
return a;
}
int numb = day + 30 * months[month];
~~for~~ vector<string> result;
for (auto event = events.find(numb); event != events.end() &&
event -> first == number; event++){
result.push_back(event -> second);
}
return result;
}

b) list<string> data = {"a", "hello", "aaaaa", ..., };

a) for (list<string>::iterator s = data.begin(); s != data.end(); data++){
    if ((*s).size() > 5){
        cout << *s << " ";
    }
}

a) for_each(data.begin(), data.end(), [](string s){ if(s.size()>5){ cout << s << " "; }});

c)
```
mutex mx;
condition_variable cv
bool isNumber = false;
char current;
string result = "";
bool finished = false;
void producer(string s){
    for(char cur: s){
        lock_guard<mutex> lk(mx);
        if(cur >= '0' && cur <= '9'){
            current = cur;
            isNumber = true;
        }else{
            current = cur;
            isNumber = false;
        }
```

c)
```
mutex mx;
condition_variable cv;
bool isNumber = false;
char current;
string result = "";
bool finished = false;

void producer(string s){
  for(char cur : s){
    {lock_guard<mutex> lk(mx);
      current = cur;
      if(cur >= '0' && cur <= '9'){
        isNumber = true
      }
      else{
        isNumber = false;
      }
      cv.notify_all();
    }
    {lock_guard<mutex> lk(mx);
      finished = true;
    }
    cv.notify_all();
}

void consumer_number(){
  while(true){
    unique_lock<mutex> lk(mx);
    cv.wait(lk, [](){return finished || isNumber;});
    if(finished) break;
    result.push_back(current);
  }
  cv.notify_all();
}

void consumer_alpha(){
  while(true){
    unique_lock<mutex> lk(mx)
    cv.wait(lk, [](){return finished || isNumber;});
    if(finished){ break;}
    result.push_back(current);
  } cv.notify_all(); }
```

```
int main(){
string source = "1a2"
thread *t1 = new thread(
    producer, source);
thread *t2 = new thread(
    consumer_number);
thread *t3 = new thread(
    consumer_alpha);
t1->join();
t2->join();
t3->join();
}
```