



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа №10
по курсу «Языки и методы программирования»
«Реализация итераторов на языке C++»

Студент группы ИУ9-22Б Лавров Р. Д.

Преподаватель Посевин Д. П.

Москва 2025

1 Задание

Выриант 13: Множество целых чисел с константным однонаправленным итератором по всем тройкам чисел, которые могут представлять длины сторон прямоугольного треугольника.

Выриант 22: Последовательность целых чисел с константным однонаправленным итератором по подпоследовательностям, сумма элементов которых не превышает 21 (подпоследовательности имеют максимально возможную длину, состоят из соседних элементов и не пересекаются).

2 Результаты

Исходный код 1–4.

Листинг 1 — NumberSet.cpp (13 вариант)

```
1 #include <iostream>
2
3 using namespace std;
4
5 class NumSet {
6 private:
7     vector<vector<int>> set;
8 public:
9     NumSet(vector<int> s) {
10         vector<vector<int>> new_set;
11         for (int i = 0; i < s.size(); i++){
12             for (int j = i+1; j < s.size(); j++){
13                 for (int k = j+1; k < s.size(); k++){
14                     auto a = s[i] * s[i];
15                     auto b = s[j] * s[j];
16                     auto c = s[k] * s[k];
17                     if (a == b + c || b == a + c || c == a + b) {
18                         vector<int> tr;
19                         tr.push_back(s[i]);
20                         tr.push_back(s[j]);
21                         tr.push_back(s[k]);
22                         new_set.push_back(tr);
23                     }
24                 }
25             }
26         }
27         set = new_set;
28     }
29
30     vector<int> operator[] (int idx) {
31         return set[idx];
32     }
```

Листинг 2 — NumberSet.cpp (13 вариант) (продолжение)

```

1  class Iterator {
2  private:
3      int idx;
4      NumSet* numSet;
5  public:
6      Iterator(NumSet* ptr, int idx) : numSet(ptr), idx(idx){}
7
8      vector<int> operator*() {
9          return (*numSet)[idx];
10     }
11
12     Iterator& operator++() {
13         ++idx;
14         return *this;
15     }
16
17     bool operator==(const Iterator& b) const{
18         return idx == b.idx;
19     }
20
21     bool operator!=(const Iterator& b) const{
22         return idx != b.idx;
23     }
24 };
25
26 Iterator begin(){ return Iterator (this, 0); }
27 Iterator end(){ return Iterator (this, set.size()); }
28 };
29
30 int main() {
31     vector<int> set1 = {3, 4, 5, 7, 12, 13};
32     NumSet numSet1(set1);
33
34     for (auto a = numSet1.begin(); a != numSet1.end(); ++a) {
35         std::cout << "(" << (*a)[0] << ", " << (*a)[1] << ", " << (*a)[2] << ")" << endl;
36     }
37 }

```

Листинг 3 — NumberSeq.cpp (22 вариант)

```

1  #include <iostream>
2
3  using namespace std;
4
5  class NumberSet {
6  private:
7      vector<vector<int>> set;
8  public:
9      NumberSet(vector<int> a) {
10         vector<vector<int>> new_set;

```

Листинг 4 — NumberSeq.cpp (22 вариант) (продолжение)

```

1      for (int i = 0; i < a.size(); i++){
2          vector<int> subsq;
3          int sum = 0;
4          for (int j = i; j < a.size(); j++){
5              if (sum + a[j] <= 21) {
6                  subsq.push_back(a[j]);
7                  sum += a[j];
8              } else {
9                  break;
10             }
11         }
12         if (subsq.size() > 0) {
13             new_set.push_back(subsq);
14         }
15     }
16     set = new_set;
17 }
18
19 vector<int> operator[](int idx) {
20     return set[idx];
21 }
22
23 class Iterator {
24 private:
25     int idx;
26     NumberSet* numSet;
27 public:
28     Iterator(NumberSet* ptr, int idx) : numSet(ptr), idx(idx){}
29     vector<int> operator*() {
30         return (*numSet)[idx];
31     }
32     Iterator& operator++() {
33         ++idx;
34         return *this;
35     }
36     bool operator==(const Iterator& b) const{
37         return idx == b.idx;
38     }
39     bool operator!=(const Iterator& b) const{
40         return idx != b.idx;
41     }
42 };
43 Iterator begin(){ return Iterator(this, 0); }
44 Iterator end(){ return Iterator(this, set.size()); }
45 };
46
47 int main() {
48     vector<int> set1 = {3, 4, 5, 43, 7, 12, 13};
49     NumberSet numSet1(set1);
50     for (auto a = numSet1.begin(); a != numSet1.end(); ++a) {
51         std::cout << "{";
52         for (int i = 0; i < (*a).size() - 1; i++){
53             std::cout << (*a)[i] << ", ";
54         }
55         std::cout << (*a)[(*a).size() - 1] << "}" << endl;
56     }
57 }

```

```
● veter.ok77@MacBook-Pro-Rodion lab10 % ./NumSet
(3, 4, 5)
(5, 12, 13)
✧ veter.ok77@MacBook-Pro-Rodion lab10 %
```

Рис. 1 — Результат работы

```
● veter.ok77@MacBook-Pro-Rodion lab10 % ./NumberSet
{3, 4, 5}
{4, 5}
{5}
{7, 12}
{12}
{13}
✧ veter.ok77@MacBook-Pro-Rodion lab10 % $
```

Рис. 2 — Результат работы

3 Вывод

Я научился делать итераторы на C++