

Задачи к лабораторной работе «Стандартная библиотека шаблонов»

Цель работы:

Изучить особенности применения контейнеров и алгоритмов стандартной библиотеки шаблонов языка C++.

Указания к выполнению работы

Для всех индивидуальных вариантов, имеющих ссылку на тестирующую систему, обязательным условием является успешное выполнение всех тестов¹.

- при решении задач 1-3 необходимо использовать как минимум 3 различных метода соответствующего контейнерного класса;
- при решении задачи 1 необходимо использовать контейнер *vector*, при этом необходимо сформировать исходный вектор по данным, вводимым пользователем, а затем обработать вектор в соответствии с условием задачи;
- при решении задачи 2 необходимо использовать контейнер *set* или *multiset*, при необходимости - тип перечисления;
- при решении задачи 3 необходимо использовать контейнер *map* или *multimap*;
- при решении задачи 4 необходимо использовать по крайней мере один из STL алгоритмов (библиотеки *algorithm*, *numeric*), с указанием лямбда-выражения.

Индивидуальные варианты заданий

Задача 1

1. Дан массив действительных чисел размера n . Удалить из массива k -й по счету элемент (k – натуральное число, $k < n$).
2. Дан массив действительных чисел размера n ($n \geq 5$, n - нечетное). Удалить три средних элемента вектора.
3. Дан массив действительных чисел размера n . Удалить из массива все элементы, начиная с $n1$ по $n2$ ($n1 < n2 \leq n$).
4. Дан массив действительных чисел размера n . Удалить из массива максимальный элемент (если таких несколько – удалить все).
5. Дан массив действительных чисел размера n . Удалить из массива элемент, имеющий минимальное по модулю значение (если таких несколько – удалить все).
6. Дан массив действительных чисел размера n . Удалить из массива первый отрицательный элемент или сообщить, что такого нет.
7. Дан массив целых чисел размера n . Вставить в массив заданное число после второго четного элемента или сообщить, что такого нет.
8. Дан массив целых чисел размера n . Удалить из массива последний четный элемент или сообщить, что такого нет.
9. Дан массив целых чисел размера n . Вставить в массив заданное число перед последним отрицательным элементом или сообщить, что такого нет.
10. Дан массив целых чисел размера n . Вставить в массив заданное число после первого нечетного элемента или сообщить, что такого нет.

¹ В этом случае прочие указания к выполнению можно не выполнять

11. Дан массив целых чисел размера n . Удалить из массива первый минимальный и последний положительный элемент или сообщить, что положительных элементов нет.
12. Дан массив целых чисел размера n . Вставить в массив два заданных числа: первое – перед максимальным, второе – после него.
13. Дан массив действительных чисел размера n . Удалить из массива все элементы, большие действительного числа k или сообщить, что таких элементов нет.
14. Дан массив целых чисел размера n . Вставить в массив заданное число после каждого элемента массива, кратного числу 5 или сообщить, что таких элементов нет.
15. Дан массив действительных чисел размера n . Удалить из массива все отрицательные элементы или сообщить, что таких элементов нет.
16. Дан массив целых чисел размера n . Вставить в массив заданное число после каждого отрицательного элемента массива или сообщить, что таких элементов нет.
17. Дан массив целых чисел размера n . Удалить из массива все кратные k элементы или сообщить, что таких элементов нет.
18. Дан массив целых чисел размера n . Удалить из массива все четные элементы, стоящие на нечетных местах или сообщить, что таких элементов нет.
19. Дан массив целых чисел размера n . Вставить в массив максимальный по модулю элемент после каждого отрицательного элемента, стоящего на четных местах или сообщить, что таких элементов нет.
20. Дан массив целых чисел размера n . Удалить из массива по одному правому соседу для каждого элемента с четным значением или сообщить, что таких элементов нет.
21. Дан массив целых чисел размера n . Удалить из массива все элементы, являющиеся простыми числами или сообщить, что таких элементов нет.
22. Дан массив целых чисел размера n . Вставить заданное пользователем значение после каждого элемента массива, содержащего простое число или сообщить, что таких элементов нет.
23. Дан массив целых чисел размера n . Вставить в массив единицу после каждого составного числа или сообщить, что таких элементов нет.
24. Дан массив целых чисел размера n . После каждого элемента, у которого есть k или более делителей вставить ноль или сообщить, что таких элементов нет.
25. Дан массив целых чисел размера n . Удалить из массива все локальные минимумы (элемент, который строго меньше значений соседних с ним ячеек) или сообщить, что таких элементов нет.
26. Дан массив целых чисел размера n . Перед каждым локальным максимумом (элемент, который строго больше значений соседних с ним ячеек) вставить минимальный по модулю элемент массива или сообщить, что таких элементов нет.
27. Дан массив целых чисел размера n . Удалить из массива все элементы, позиции которых являются числами Фибоначчи.
28. Дан массив целых чисел размера n . Перед каждым значением, которое является числом Люка вставить ноль или сообщить, что таких элементов нет.
29. Дан массив целых чисел размера n . Перед первым минимальным элементом массива вставить все его делители.
30. Дан массив целых чисел размера n . Вставить перед каждым четным по значению элементом массива все числа Фибоначчи, не превышающие текущее значение или сообщить, что таких элементов нет.
31. https://acmp.ru/index.asp?main=task&id_task=293
32. https://acmp.ru/index.asp?main=task&id_task=642
33. https://acmp.ru/index.asp?main=task&id_task=9
34. https://acmp.ru/index.asp?main=task&id_task=314
35. <https://codeforces.com/gym/101911/problem/K>
36. https://acmp.ru/index.asp?main=task&id_task=472

Задача 2

1. Дан массив целых чисел размера n (n – четное). Если все значения, содержащиеся во второй половине массива, входят хотя бы один раз в его первую половину, то вывести *true*, иначе вывести *false*.
2. Дана строка из строчных латинских букв. Напечатать в алфавитном порядке все буквы, входящие в текст по одному разу.
3. Дана строка из строчных латинских букв. Напечатать в алфавитном порядке все буквы, входящие в текст не менее двух раз.
4. Дан текст, за которым следует точка. В алфавитном порядке напечатать все строчные латинские гласные буквы (*a, e, i, o, u, y*), входящие в этот текст.
5. Дан текст из цифр и строчных латинских букв, за которыми следует точка. Определить, каких букв — гласных (*a, e, i, o, u, y*) или согласных — больше в этом тексте.
6. Задано некоторое множество $m1$ и множество $m2$ того же типа. Подсчитать, сколько элементов из множеств $m1$ и $m2$ совпадает.
7. Составить программу, которая вычисляет сумму тех элементов двумерного массива, номера строк и столбцов которых принадлежат соответственно непустым множествам $m1$ и $m2$.
8. Подсчитать количество различных цифр в десятичной записи натурального числа.
9. Вводится натуральное число. Выписать в возрастающем порядке все цифры, не входящие в запись данного числа.
10. Имеется множество, содержащее натуральные числа из некоторого диапазона. Сформировать два множества, первое из которых содержит все простые числа из данного множества, а второе — все составные.
11. Дан массив из n целых чисел ($n \geq 3$). Вывести все его различные элементы, кроме максимального и минимального, в порядке убывания.
12. Дан массив из n целых чисел ($n \geq 3$). Вывести все его элементы (с учетом повторений), кроме максимального и минимального, в порядке возрастания.
13. Дан текст на английском языке. Напечатать в алфавитном порядке все гласные буквы (*a, e, i, o, u, y*), которые входят в каждое слово.
14. Дан текст на английском языке. Напечатать в алфавитном порядке все согласные буквы, которые входят более, чем в одно слово.
15. Дан текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые не входят хотя бы в одно слово.
16. Дан текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые входят в каждое нечетное слово и не входят хотя бы в одно четное слово.
17. Дан текст на русском языке. Напечатать в алфавитном порядке все звонкие согласные буквы, которые входят в каждое нечетное слово и не входят ни в одно четное слово.
18. Дан массив из n натуральных чисел. Выписать в возрастающем порядке те цифры, которые не входят ни в одно из чисел массива.
19. Дан массив из n натуральных чисел. Выписать в возрастающем порядке те цифры, которые входят во все числа массива.
20. Дан массив из n натуральных чисел. Выписать в убывающем порядке те цифры, которые не входят ни в одно из четных чисел массива.
21. Дан массив из n натуральных чисел. Выписать в убывающем порядке те цифры, которые входят во все нечетные числа.

22. Дан массив из n целых чисел (n – четное). Вывести на экран в возрастающем порядке различные числа, которые одновременно входят и в первую, и во вторую половину массива.
23. Дан массив из n целых чисел (n – четное). Вывести на экран в убывающем порядке различные числа, которые одновременно входят во вторую половину массива и при этом отсутствуют в первой половине.
24. Дан массив из n целых чисел (n – четное). Вывести на экран в возрастающем порядке числа (с учетом повторений), которые входят в первую половину массива, и при этом отсутствуют во второй его половине.
25. Из диапазона целых чисел $a...b$ выделить два множества: 1-е множество – из чисел, делящихся без остатка на k или на t (k, t — простые), 2-е множество – из чисел, делящихся без остатка на k и на t одновременно.
26. Дано натуральное число n . Используя метод «решета Эратосфена» выписать все простые числа от 1 до n в убывающем порядке.
27. Дан текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые не входят только в одно слово.
28. Дан текст на русском языке. Напечатать в алфавитном порядке все звонкие согласные буквы, которые входят более чем в одно слово.
29. Даны массивы u и v целых чисел размеров n и m соответственно. Найти все числа (с учетом повторений), которые входят хотя бы в один из исходных массивов, и вывести их в порядке возрастания; при этом если, например, некоторое число входит в один из массивов 3 раза, а в другой 5 раз, то его надо вывести 5 раз.
30. Даны массивы u и v целых чисел размеров n и m соответственно. Найти все числа (с учетом повторений), которые входят в один из исходных массивов и отсутствуют в другом, и вывести их в порядке убывания; при этом если, например, некоторое число входит в один из векторов 3 раза, а в другой 5 раз, то его надо вывести 2 раза.
31. https://acmp.ru/index.asp?main=task&id_task=10
32. https://acmp.ru/index.asp?main=task&id_task=769
33. <https://codeforces.com/gym/101911/problem/C>
34. https://acmp.ru/index.asp?main=task&id_task=32
35. <https://codeforces.com/problemset/problem/205/D?locale=ru>
36. https://acmp.ru/index.asp?main=task&id_task=945
37. <https://codeforces.com/problemset/problem/1154/E?locale=ru>

Задача 3

1. Дана строка s , состоящая из строчных латинских букв. Определить количество повторений каждой буквы в строке s и вывести все различные буквы строки s вместе с количеством их повторений (в порядке возрастания значений элементов); количество повторений выводить сразу после значения соответствующего элемента. Использовать вспомогательное отображение m , ключами которого являются различные буквы строки s , а значениями – количество их повторений. При заполнении отображения m не использовать условные конструкции.
2. Дан натуральное число n . Определить количество вхождений в число n каждой из его цифр. Вывести все различные цифры n вместе с количеством их повторений (в порядке возрастания цифр); количество повторений выводить сразу после значения цифры. Использовать вспомогательное отображение m , ключами которого являются различные цифры числа n , а значениями – количество повторений этих цифр. При заполнении отображения m не использовать условные конструкции.
3. Дан вектор v , элементами которого являются целые числа, все элементы вектора различны. Определить количество повторений каждого числа в векторе v и вывести все различные элементы вектора v вместе с количеством их повторений (в порядке возрастания значений элементов); количество повторений выводить сразу после

значения соответствующего элемента. Использовать вспомогательное отображение m , ключами которого являются различные элементы вектора v , а значениями — количество повторений этих элементов. При заполнении отображения m не использовать условные конструкции.

4. Дан текст из нескольких предложений. Определить количество повторений каждого из знаков пунктуации в тексте и вывести все различные знаки пунктуации вместе с количеством их повторений (в порядке возрастания кодов символов); количество повторений выводить сразу после значения соответствующего элемента. Использовать вспомогательное отображение m , ключами которого являются различные знаки пунктуации, а значениями — количество повторений этих элементов. При заполнении отображения m не использовать условные конструкции.

5. Дан вектор v , элементами которого являются английские слова, набранные строчными буквами. Для каждой буквы латинского алфавита определить количество слов вектора v , в которые она входит. Вывести все различные буквы (в алфавитном порядке), и количество слов, в которые входит указанная буква. Использовать вспомогательное отображение m , ключами которого являются буквы английского алфавита, а значениями — количество слов вектора v , в которые входит данная буква. При заполнении отображения m не использовать условные конструкции.

6. Дан вектор v , элементами которого являются английские слова, набранные строчными буквами. Определить количество слов, начинающихся на каждую из всех первых букв слов вектора v . Вывести все различные буквы, и количество слов, которое начинается на указанную букву (в алфавитном порядке букв). Использовать вспомогательное отображение m , ключами которого являются начальные буквы элементов вектора v , а значениями — количество слов, начинающихся на данную букву. При заполнении отображения m не использовать условные конструкции.

7. Дан текст из нескольких предложений. Частотная таблица представляет собой список слов и количество их вхождений в текст. Получить отображение m , содержащее частотную таблицу слов текста.

8. Дан вектор v , элементами которого являются английские слова, набранные заглавными буквами. Определить суммарную длину слов, начинающихся с одной и той же буквы, и вывести все различные буквы, с которых начинаются элементы вектора v , вместе с суммарной длиной этих элементов (в алфавитном порядке букв); длину выводить сразу после соответствующей буквы. Использовать вспомогательное отображение m , ключами которого являются начальные буквы элементов вектора v , а значениями — суммарная длина этих элементов. При заполнении отображения m не использовать условные конструкции.

9. Дан вектор v с элементами типа $pair<string, float>$. В каждой группе его элементов с одинаковым строковым значением, найти сумму действительных значений этих элементов. Для каждой группы вывести соответствующую ей строку и найденную сумму, упорядочивая выводимые пары в соответствии с алфавитным порядком строковых значений. Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, а значениями — полученные суммы (таким образом, отображение m должно иметь тип $map<string, float>$). При заполнении отображения m не использовать условные конструкции.

10. Дан вектор v , элементами которого являются целые числа. В каждой группе его элементов, оканчивающихся на одну и ту же цифру, найти сумму значений этих элементов, за исключением начального элемента группы (предполагается, что элементы группы располагаются в том же порядке, что и в исходном векторе). Если группа состоит из единственного элемента, то сумма должна равняться 0. Для каждой группы вывести соответствующую ей цифру и найденную сумму, упорядочивая выводимые пары по возрастанию цифр. Представить результат группировки в виде

мультиотображения m , ключами которого являются ключи группировки, а значениями — полученные суммы (таким образом, m должно иметь тип `multimap<int, int>`).

11. Дан вектор v , элементами которого являются целые числа, все элементы вектора различны. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки последнюю (т. е. правую) цифру элемента: в одну группу должны входить все элементы вектора v , оканчивающиеся на одну и ту же цифру (сгруппированные элементы должны располагаться в том же порядке, в котором они располагались в исходном векторе). Представить результат группировки в виде мультиотображения m , ключами которого являются ключи группировки, а значениями — векторы, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип `multimap<int, vector<int>>`). Вывести полученное отображение (для каждого элемента мультиотображения m вначале выводить ключ, а затем элементы связанного с ним вектора; ключи могут повторяться).

12. Дан вектор v , элементами которого являются целые числа; числа в векторе могут повторяться. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки первую (т. е. левую) цифру элемента: в одну группу должны входить все элементы вектора v , начинающиеся на одну и ту же цифру (сгруппированные элементы должны располагаться в порядке же порядке возрастания их значений). Представить результат группировки в виде мультиотображения m , ключами которого являются ключи группировки, т. е. первые цифры элементов вектора v , а значениями — элементы вектора, начинающиеся на соответствующую цифру (таким образом, m должно иметь тип `multimap<int, multiset<int>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем связанный с ним элемент вектора v ; ключи могут повторяться).

13. Дан вектор v , элементами которого являются английские слова, набранные заглавными буквами. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки длину слова: в одну группу должны входить все элементы вектора v с одинаковой длиной (сгруппированные элементы должны располагаться в том же порядке, в котором они располагались в исходном векторе). Представить результат группировки в виде отображения m , ключами которого являются ключ группировки, а значениями — векторы, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип `map<int, vector<string>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним вектора).

14. Дан вектор v , элементами которого являются английские слова, набранные строчными буквами. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки количество слогов в слове: в одну группу должны входить все элементы вектора v с одинаковым числом слогов (сгруппированные элементы должны располагаться в том же порядке, в котором они располагались в исходном векторе). Представить результат группировки в виде отображения m , ключами которого являются ключ группировки, а значениями — векторы, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип `map<int, vector<string>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним вектора).

15. Дан вектор v , каждый элемент которого является текстом на английском языке, предложения в котором разделены точкой. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки количество предложений в тексте: в одну группу должны входить все элементы вектора v с одинаковым числом предложений (сгруппированные элементы должны располагаться в том же порядке, в котором они располагались в исходном векторе). Представить результат группировки в виде отображения m , ключами которого являются ключ группировки, а значениями — векторы, содержащие сгруппированные элементы (таким образом, отображение m

должно иметь тип `map<int, vector<string>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним вектора).

16. Дан вектор v , элементами которого являются целые числа, элементы вектора не повторяются. Выполнить группировку элементов вектора v , используя в качестве ключа группировки количество цифр элемента вектора v : в одну группу должны входить все целочисленные элементы, с одинаковым количеством цифр (сгруппированные элементы должны располагаться в порядке, обратном порядку их расположения в исходном векторе). Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, а значениями — векторы, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип `map<int, vector<int>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним вектора).

17. Дан вектор v , элементами которого являются английские слова, набранные заглавными буквами. Выполнить группировку элементов вектора v , используя в качестве ключа группировки первую букву элемента: в одну группу должны входить все элементы вектора v , начинающиеся с одной и той же буквы (сгруппированные элементы должны располагаться в алфавитном порядке с учетом повторений). Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, а значениями — мультимножества, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип `map<char, multiset<string>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним мультимножества).

18. Дан вектор v , элементами которого являются английские слова, набранные заглавными буквами. Выполнить группировку элементов вектора v , используя в качестве ключа группировки последнюю букву элемента: в одну группу должны входить все элементы вектора v , оканчивающиеся одной и той же буквой (сгруппированные элементы должны располагаться в порядке, обратном порядку их расположения в исходном векторе). Представить результат группировки в виде мультиотображения m , ключами которого являются ключи группировки, т. е. последние буквы элементов вектора v , а значениями — элементы вектора, оканчивающиеся на соответствующую букву (таким образом, m должно иметь тип `multimap<char, vector<string>>`). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем связанный с ним элемент вектора v ; ключи могут повторяться).

19. Дан вектор v , элементами которого являются английские слова, набранные заглавными буквами. В каждой группе его элементов, оканчивающихся на одну и ту же букву, получить строку, являющуюся суммой всех слов из этой группы, кроме последнего слова (предполагается, что элементы группы располагаются в том же порядке, что и в исходном векторе). При построении строки добавлять после каждого слова пробел. Если группа состоит из единственного элемента, то строка должна остаться пустой. Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, т. е. последние буквы элементов вектора v , а значениями — суммой всех слов из этой группы (таким образом, m должно иметь тип `map<char, string>`). Для каждой группы вывести соответствующую ей букву и найденную строку, упорядочивая выводимые пары в алфавитном порядке букв.

20. Дан вектор v , элементами которого являются английские слова, набранные заглавными буквами. Выполнить группировку элементов вектора v , используя в качестве ключа группировки все различные буквы слов: в одну группу должны входить все элементы вектора v , содержащие букву ключа (сгруппированные элементы должны располагаться в алфавитном порядке). Представить результат группировки в виде

отображения m , ключами которого являются ключи группировки, т. е. все различные буквы слов вектора v , а значениями — элементы вектора, содержащие соответствующую букву (таким образом, m должно иметь тип $\text{map}<\text{char}, \text{multiset}<\text{string}>>$). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем связанные с ним элементы вектора v).

21. Дан вектор v с элементами типа $\text{pair}<\text{char}, \text{int}>$. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки символьный элемент (значения символа в разных парах могут совпадать): в одну группу должны входить все целочисленные элементы, с одинаковым символьным значением (сгруппированные элементы должны располагаться в порядке, обратном порядку их расположения в исходном векторе). Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, а значениями — вектора, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип $\text{map}<\text{char}, \text{vector}<\text{int}>>$). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним вектора).

22. Дан вектор v с элементами типа $\text{pair}<\text{string}, \text{int}>$. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки строковый элемент (значения строк в разных парах могут совпадать): в одну группу должны входить все целочисленные элементы, с одинаковым строковым значением (сгруппированные элементы должны располагаться в порядке возрастания). Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, а значениями — мультимножества, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип $\text{map}<\text{string}, \text{multiset}<\text{int}>>$). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним мультимножества).

23. Дан вектор v , элементами которого являются целые числа, элементы вектора не повторяются. Выполнить *группировку* элементов вектора v , используя в качестве ключа группировки количество делителей целого числа: в одну группу должны входить все целочисленные элементы, с одинаковым количеством делителей (сгруппированные элементы должны располагаться в порядке возрастания). Представить результат группировки в виде отображения m , ключами которого являются ключи группировки, а значениями — векторы, содержащие сгруппированные элементы (таким образом, отображение m должно иметь тип $\text{map}<\text{int}, \text{set}<\text{int}>>$). Вывести полученное отображение (для каждого элемента отображения m вначале выводить ключ, а затем элементы связанного с ним вектора).

24. Даны векторы $v1$ и $v2$, элементами которых являются целые числа; все элементы в каждом векторе различны. Получить вектор v , содержащий пары чисел (типа $\text{pair}<\text{int}, \text{int}>$), удовлетворяющие следующим условиям: первый элемент пары принадлежит вектору $v1$, второй принадлежит вектору $v2$, и оба элемента оканчиваются одной и той же цифрой. Полученный набор пар называется *внутренним объединением* векторов $v1$ и $v2$, по ключу, определяемому последними цифрами исходных чисел. Порядок следования пар определяется исходным порядком элементов вектора $v1$, а для равных первых элементов — исходным порядком элементов вектора $v2$. Для построения вектора v выполнить группировку элементов вектора $v2$ по ключу — последней цифре.

25. Даны векторы $v1$ и $v2$, элементами которых являются английские слова, набранные заглавными буквами, причем все слова в каждом векторе различны. Получить вектор v , являющийся внутренним объединением векторов $v1$ и $v2$ (см. задачу 24), каждая пара которого содержит слова одинаковой длины. Порядок следования пар определяется алфавитным порядком первых элементов пар, а для равных первых элементов — порядком вторых элементов, обратным алфавитному. Для построения вектора v выполнить группировку элементов вектора $v2$ по ключу — длине слова, используя вариант группировки со вспомогательным отображением m типа

map<int, set<string>>, после чего в цикле по отсортированным элементам вектора *v1* сформировать требуемое внутреннее объединение, перебирая для каждого элемента вектора *v1* соответствующие ему элементы отображения *m*.

26. Даны векторы *v1* и *v2*, элементами которых являются английские слова, набранные заглавными буквами, причем все слова в каждом векторе различны. Получить вектор *v*, являющийся внутренним объединением векторов *v1* и *v2* (см. задачу 24), в каждой паре которого первое слово начинается с буквы, которой оканчивается второе слово. Порядок следования пар определяется алфавитным порядком первых элементов пар, а для равных первых элементов — порядком вторых элементов, обратным порядку их следования в векторе *v2*. Для построения вектора *v* выполнить группировку элементов вектора *v2* по ключу — последней букве слова, используя вариант группировки со вспомогательным мультиотображением *m*, после чего в цикле по отсортированным элементам вектора *v1* сформировать требуемое внутреннее объединение, перебирая для каждого элемента вектора *v1* соответствующие ему элементы отображения *m*.

27. Даны векторы *v1* и *v2*, все элементы в каждом векторе различны и являются положительными числами. Получить вектор *v* пар типа *pair<int, vector<int>>*, в которых первый элемент пары принадлежит вектору *v1*, а второй представляет собой набор тех элементов вектора *v2*, которые оканчиваются той же цифрой, что и первый элемент пары (если подходящие элементы в векторе *v2* отсутствуют, то второй элемент пары должен содержать единственный элемент, равный 0). Полученный набор пар называется *левым внешним объединением* векторов *v1* и *v2* по ключу, определяемому последними цифрами исходных чисел. Порядок следования пар определяется исходным порядком элементов вектора *v1*; порядок чисел, входящих во вторые элементы пар, определяется исходным порядком элементов вектора *v2*. Для построения вектора *v* выполнить группировку элементов вектора *v2* по ключу — последней цифре, используя вариант группировки со вспомогательным отображением *m*, после чего в цикле по элементам вектора *v1* сформировать требуемое внешнее объединение, перебирая для каждого элемента вектора *v1* соответствующие ему элементы отображения *m*. Вывести полученный вектор *v*, указывая после первого элемента пары все числа, входящие во второй элемент данной пары.

28. Даны векторы *v1* и *v2*, элементами которых являются английские слова, набранные заглавными буквами, причем все слова в каждом векторе различны. Получить левое внешнее объединение векторов *v1* и *v2* (см. задачу 27) по ключу — длине слова: каждому элементу вектора *v1* должен соответствовать набор всех элементов вектора *v2*, имеющих ту же длину, или набор, содержащий только пустую строку, если требуемые элементы в векторе *v2* отсутствуют. Левое внешнее объединение должно быть упорядочено в алфавитном порядке элементов из вектора *v1*; в каждом наборе элементов, соответствующих элементу из *v1*, порядок должен быть обратным алфавитному порядку. Реализовать левое внешнее объединение в виде отображения *m* с ключом — строкой и значением — строковым множеством. Для построения множества *m* выполнить группировку элементов вектора *v2* по ключу — длине слова, используя вариант группировки со вспомогательным отображением *m0* типа *map<int, set<string, greater<string>>>*, после чего в цикле по элементам вектора *v1* сформировать требуемое внутреннее объединение, перебирая для каждого элемента вектора *v1* соответствующие ему элементы отображения *m0*. Вывести полученное отображение *m*, указывая после каждого его ключа (т. е. элемента вектора *v1*) все слова, входящие в связанное с этим ключом множество значений (т. е. все соответствующие ему элементы вектора *v2* или, при их отсутствии, пустую строку).

29. Даны векторы *v1* и *v2*; все элементы в каждом векторе различны и являются положительными числами. Получить вектор *v* пар типа *pair<int, int>*, в которых первый элемент пары принадлежит вектору *v1*, а второй представляет собой

сумму значений тех элементов вектора v_2 , которые оканчиваются той же цифрой, что и первый элемент пары (если подходящие элементы в векторе v_2 отсутствуют, то второй элемент пары должен быть равен 0). Порядок следования пар должен быть обратным порядку следования элементов вектора v_1 . При построении вектора v использовать вспомогательное отображение m , построенное на основе вектора v_2 и сопоставляющее каждой цифре сумму элементов вектора v_2 , оканчивающихся этой цифрой.

30. Даны векторы v_1 и v_2 , элементами которых являются английские слова, набранные заглавными буквами, причем все слова в каждом векторе различны. Получить отображение m , в котором ключом является элемент вектора v_1 , а значением — строка, полученная суммированием следующего набора элементов вектора v_2 : в набор включаются все слова, оканчивающиеся на ту же букву, что и ключ, за исключением последнего подходящего слова. Порядок слов в наборе должен соответствовать их порядку в векторе v_2 . При построении строки добавлять после каждого слова пробел. Если требуемый набор является пустым, то строка также должна остаться пустой. При построении отображения m использовать вспомогательное отображение m_0 , построенное на основе вектора v_2 и сопоставляющее каждой букве описанную выше сумму элементов вектора v_2 , оканчивающихся этой буквой.

31. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=12&id_problem=592

32. https://acmp.ru/index.asp?main=task&id_task=984

33. https://acmp.ru/index.asp?main=task&id_task=997

34. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=12&id_problem=1194

35. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=12&id_problem=1180

36. https://acmp.ru/index.asp?main=task&id_task=964

37. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=12&id_problem=69

Задача 4

1. Дан вектор v . Удалить второй из элементов вектора, равных нулю. Если нулевых элементов меньше двух, то вектор не изменять. Использовать два вызова алгоритма *find* и функцию-член *erase*.

2. Дан дек d . Удалить последний нулевой элемент дека. Если нулевых элементов нет, то дек не изменять. Использовать алгоритм *find* с обратными итераторами и функцию-член *erase*. Указание: поскольку функция-член *erase* не позволяет использовать обратный итератор для удаления элемента, необходимо перейти от обратного итератора к связанному с ним обычному итератору, используя функцию-член обратного итератора *base*.

3. Дан список l . Удалить первый и последний нулевой элемент списка. Если нулевых элементов нет, то список не изменять, если нулевой элемент всего один, то удалить только его. Использовать два вызова алгоритма *find* и два вызова функции-члена *erase*.

4. Дан список l . Удалить все элементы списка, расположенные между первым и вторым отрицательным элементом (не включая сами отрицательные элементы). Если список не содержит отрицательных элементов, то не изменять его, если отрицательный элемент всего один, то удалить все элементы, следующие за этим отрицательным элементом. Использовать два вызова алгоритма *find_if* и один вызов функции-члена *erase*.

5. Дан список l , содержащий как отрицательные, так и положительные элементы. Вставить нулевой элемент после первого отрицательного элемента и перед

последним положительным элементом. Использовать два вызова алгоритма *find_if* и два вызова функции-члена *insert*.

6. Даны вектор v и список l ; вектор v имеет четное количество элементов. Продублировать последний элемент списка, совпадающий с каким-либо элементом из первой половины исходного вектора. Если список не содержит требуемых элементов, то не изменять его. Использовать алгоритм *find_first_of* и функцию-член *insert* для списка.

7. Дан вектор v с четным количеством элементов. Добавить нулевой элемент перед последним элементом в первой половине вектора, совпадающим с каким-либо элементом из второй половины этого же вектора. Если вектор не содержит требуемых элементов, то не изменять его. Использовать алгоритм *find_first_of* и функцию-член *insert*.

8. Дано целое число k ($k > 0$) и вектор v , содержащий только нули и единицы. Удалить в векторе v последний набор из k подряд расположенных нулей (если в этом наборе имеется больше k нулей, то требуется удалить только последние k из них). Если вектор не содержит требуемого набора нулей, то не изменять его. Использовать алгоритм *search_n* и функцию-член *erase*, а также обратные итераторы.

9. Даны список l и дек d ; дек d содержит четное количество элементов. Продублировать в деке d первый набор элементов, расположенный в его второй половине и совпадающий с элементами списка l , взятыми в обратном порядке. Например, для списка $\{1, 2\}$ и дека $\{2, 1, 1, 2, 3, 2, 1, 1, 2, 1\}$ дек должен быть преобразован следующим образом: $\{2, 1, 1, 2, 3, 2, 1, (2, 1), 1, 2, 1\}$ (вставленные элементы заключены в скобки). Если дек не содержит требуемого набора чисел, то не изменять его. Использовать алгоритм *search* и функцию-член *insert*.

10. Дан вектор v с четным количеством элементов. Удалить все нулевые элементы, расположенные во второй половине исходного вектора. Использовать алгоритм *remove* и функцию-член *erase*. *Указание.* Алгоритм *remove* не удаляет требуемые элементы, а лишь перемещают их в конец указанного диапазона и возвращает итератор на начало диапазона с перемещенными элементами. Для фактического удаления элементов после выполнения алгоритма необходимо вызвать функцию-член *erase*.

11. Дан список l с четным количеством элементов n . Добавить в начало списка $n/2$ новых элементов со следующими значениями: $a_1 + a_n, a_2 + a_{n-1}, \dots, a_{n/2} + a_{n/2+1}$, где a_1, a_2, \dots, a_n обозначают исходные элементы списка. Использовать алгоритм *transform* с обратным итератором и итератором вставки.

12. Дан дек d с четным количеством элементов. Заменить в нем вторую половину элементов на удвоенные значения элементов первой половины, расположив эти удвоенные значения в обратном порядке. Использовать алгоритм *transform* с обратным итератором.

13. Дан вектор v с четным количеством элементов. Найти сумму элементов из первой и из второй половины вектора. Использовать два вызова алгоритма *accumulate*.

14. Даны объекты a и b и векторы $v1$ и $v2$, каждый из которых содержит не менее 10 элементов. Заполнить первые 5 элементов каждого вектора значениями a , а последние 5 — значениями b . При преобразовании вектора $v1$ использовать два вызова алгоритма *fill*, при преобразовании вектора $v2$ использовать два вызова алгоритма *fill_n*.

15. Даны объекты a и b и векторы $v1$ и $v2$. Добавить в начало каждого вектора 5 элементов со значениями a , а в конец — 5 элементов со значениями b . При преобразовании вектора $v1$ использовать два вызова алгоритма *fill_n* с функциями *inserter* и *back_inserter*, при преобразовании вектора $v2$ использовать два вызова функции-члена *insert*.

16. Даны списки $l1$ и $l2$, каждый из которых содержит четное количество элементов. Поменять местами первую и вторую половину каждого списка (например, список $\{1, 2, 3, 4\}$ должен быть преобразован следующим образом: $\{3, 4, 1, 2\}$). Для

первого списка использовать алгоритм *swap_ranges*, для второго — алгоритм *rotate*. Использовать также функцию *advance*.

17. Дано число k ($0 < k < 10$) и списки $l1$ и $l2$, каждый из которых содержит не менее 10 элементов. Выполнить для списка $l1$ циклический сдвиг элементов вправо на k позиций, а для списка $l2$ — циклический сдвиг влево на k позиций. Использовать алгоритм *rotate* и функцию *advance*.

18. Дано число k ($0 < k < 5$) и список l , содержащий не менее 10 элементов. Набор из первых 5 элементов списка скопировать в конец списка, выполнив для этой копии циклический сдвиг на k позиций вправо, а набор из последних 5 элементов исходного списка скопировать в начало списка, выполнив для этой копии циклический сдвиг на k позиций влево. Использовать два вызова алгоритма *rotate_copy* и итераторы вставки, а также функцию *advance*.

19. Даны списки $l1$ и $l2$, у каждого из которых количество элементов делится на 4. В первом списке инвертировать (расположить в обратном порядке) первую половину элементов, во втором списке — вторую половину. Для первого списка использовать алгоритм *swap_ranges* и обратные итераторы, для второго — алгоритм *reverse*. Использовать также функцию *advance*.

20. Дан список l . Удвоить значения в последней паре соседних совпадающих элементов. Например, список $\{1, 2, 2, 3, 3, 1, 2, 2, 2, 4\}$ должен быть преобразован следующим образом: $\{1, 2, 2, 3, 3, 1, 2, 4, 4, 4\}$. Если список не содержит соседних совпадающих элементов, то не изменять его. Использовать алгоритм *adjacent_find* и обратные итераторы.

21. Даны списки $l1$ и $l2$, каждый из которых содержит четное количество элементов. В каждом списке продублировать первую половину, добавив ее элементы в конец списка в обратном порядке. Для первого списка использовать алгоритм *reverse_copy* и итератор вставки, для второго — функцию-член *insert* и обратные итераторы. Использовать также функцию *advance*.

22. Дан вектор v с нечетным количеством элементов n ($n \geq 3$). Определить значения трех средних элементов вектора после того, как вектор будет отсортирован (по возрастанию), и вывести их в порядке возрастания. Использовать алгоритмы *nth_element* (для нахождения центрального элемента), *max_element* (для нахождения элемента, предшествующего центральному) и *min_element* (для нахождения элемента, следующего за центральным).

23. Дан вектор v с четным количеством элементов. Удалить все четные элементы, расположенные в первой половине исходного вектора. Использовать алгоритм *remove_if* и функцию-член *erase*.

24. Даны вектор v и дек d ; дек d содержит четное количество элементов. Удалить в векторе v последний набор элементов, в котором четные и нечетные числа располагаются в том же порядке, что и в первой половине дека d . Например, для дека $\{1, 2, 3, 4, 5, 6\}$ и вектора $\{11, 14, (15, 16, 17,) 17, 18, 10\}$ в векторе должны быть удалены элементы, заключенные в скобки. Если вектор не содержит требуемого набора чисел, то не изменять его. Использовать алгоритм *search* с параметром — функциональным объектом и функцию-член *erase*.

25. Дано целое число k ($k > 0$) и вектор v . Продублировать в векторе v первый набор из k подряд расположенных положительных чисел, вставив после этого набора его копию (если в наборе имеется больше k положительных чисел, то лишние числа во внимание не принимаются, и дублируется только начальная часть набора, содержащая k чисел). Если вектор не содержит требуемого набора чисел, то не изменять его. Использовать алгоритм *search_n* с параметром — функциональным объектом.

26. Дан вектор v с четным количеством элементов. В первой половине исходного вектора заменить все отрицательные числа на -1 , а во второй — все

положительные числа на 1. Использовать два вызова алгоритма *replace_if* с различными параметрами — функциональными объектами.

27. Дан дек d , элементами которого являются английские слова. Отсортировать его элементы по убыванию их длин, а элементы одинаковой длины — в алфавитном порядке. Использовать единственный вызов алгоритма *sort* с параметром — функциональным объектом, включающим как сравнение строк, так и сравнение их длин.

28. Дан вектор v . Найти сумму отрицательных и сумму положительных элементов вектора. Использовать два вызова алгоритма *accumulate* с параметрами — функциональными объектами.

29. Дан вектор v , элементами которого являются английские слова. Получить строку, содержащую начальные символы всех элементов вектора, расположенные в обратном порядке. Использовать алгоритм *accumulate* с параметром — функциональным объектом.

30. Дан вектор v , элементами которого являются английские слова. Получить строку, содержащую конечные символы всех элементов вектора, расположенные в исходном порядке. Использовать алгоритм *accumulate* с параметром — функциональным объектом.

31. https://acmp.ru/index.asp?main=task&id_task=693

32. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=13&id_problem=72

33. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=14&id_problem=1144

34. <https://codeforces.com/gym/102375/problem/D>

35. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=14&id_problem=630

36. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=14&id_problem=78

37. https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=14&id_topic=13&id_problem=71