

Лабораторная работа 2.

Стандартная библиотека шаблонов. Контейнеры `vector`, `string`.

Лабораторная работа состоит из двух частей:

Часть 1. Набора задач, который необходимо выполнить в системе с автоматической проверкой решения. Каждая задача стоит один балл. Задача считается выполненной если у нее статус **ОК** в посылке.

Часть 2. Индивидуального или уникального задания, непосредственно связанного с лабораторной работой.

Все лабораторные работы сдаются очно, мною может быть проверена любая задача из части 1. Задания части 2 проверяются полностью.

Часть 1.

Выполните следующие задачи по ссылке:

<https://informatics.msk.ru/mod/statements/view.php?id=80883#1>

Часть 2.

Задача 1.

Напишите функцию `max_element`, которая принимает на вход матрицу и возвращает индексы (номера строки и столбца) первого вхождения максимального элемента при рассмотрении элементов матрицы построчно. Заголовок функции должен быть таким:

```
std::pair<size_t, size_t> max_element(const std::vector<std::vector<int>>& matrix)
```

Пример входных и выходных данных.

Ввод	Вывод
3 4 0 3 2 4 2 3 5 5 5 1 2 3	1 2
1 1 1	0 0
3 5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	2 4

Задача 2.

Напишите функцию `std::string common_suffix(const std::string& a, const std::string& b)`, возвращающую наибольший общий суффикс двух данных строк.

(Суффиксом строки называется всякая подстрока этой строки, на которую исходная строка заканчивается. Например, суффиксы слова `apple` — пустая строка, `e`, `le`, `ple`, `rple`, `apple`. Для слов `apple` и `tuple` наибольшим общим суффиксом является `ple`.)

Задача 3.

Дана строка, возможно, содержащая пробелы. Извлеките из этой строки все символы, являющиеся цифрами, и составьте из них новую строку. Решение оформите в виде функции

`std::string ExtractDigits(const std::string& s)`

Ввод	Вывод
<code>2+2=4</code>	<code>224</code>
<code>0123456789 less then 9876543210</code>	<code>01234567899876543210</code>
<code>This is a number: 809!</code>	<code>809</code>

Задача 4.

Вам требуется написать на C++ функцию со следующим заголовком:

`std::vector<std::string> Split(const std::string& str, char delimiter);`

Функция должна вернуть вектор строк, полученный разбиением строки `str` на части по указанному символу-разделителю `delimiter`. Если разделитель встретился в начале или в конце строки, то в начале (соответственно, в конце) вектора с результатом должна быть пустая строка. Если два разделителя встретились рядом, то пустая строка между ними тоже должна попасть в ответ. Для пустой строки надо вернуть пустой вектор

Ввод	Вывод
<code>What_is_your_name?</code> —	<code>What</code> <code>is</code> <code>your</code> <code>name?</code>
<code>13-24--08-11-2014</code>	<code>13</code> <code>24</code> <code>08</code> <code>11</code> <code>2014</code>

Задача 5.

Вам требуется написать функцию Join со следующим заголовком:

```
std::string Join(const std::vector<std::string>& tokens, char delimiter);
```

Функция должна вернуть строку, полученную склейкой элементов вектора через указанный разделитель.

Например, Join({"What", "is", "your", "name?"}, '_') должна вернуть строку "What_is_your_name?".

Задача 6.

Определите структуру "Студент" с полями имя, фамилия, дата, месяц и год рождения. Создайте вектор из таких структур, заполните его из входных данных, отсортируйте по возрастанию даты рождения либо по фамилии и имени в лексикографическом порядке и выведите отсортированный список.

При сортировке по дате рождения при равенстве года, месяца и числа рождения, сравнивайте лексикографически фамилии и имена.

При сортировке по фамилии и имени, сравнивайте сначала фамилии, затем имена при равенстве - даты рождения.

Формат ввода

Первая строка содержит одно целое число N от 0 до 10^4 - число студентов.

Далее идут N строк, каждая из которых содержит две строки длиной от 1 до 15 символов - имя и фамилию очередного студента, и три целых числа от 0 до 10^9 - день, месяц и год рождения.

Следующая строка содержит способ сортировки - "name" или "date".

Формат вывода

Выведите список студентов в формате "имя фамилия день.месяц.год".

Ввод	Вывод
3 Ivan Ivanov 1 1 1901 Petr Petrov 2 2 1902 Alexander Sidorov 3 3 1903 date	Ivan Ivanov 1.1.1901 Petr Petrov 2.2.1902 Alexander Sidorov 3.3.1903

Задача 7. (Задание с индивидуальным вариантом)

Имеется последовательность произвольных целых ненулевых чисел, в интервале от -3200 до +3200, завершающаяся числом 0, для генерации случайных чисел использовать встроенный алгоритм Вихря Мерсенна. Последовательность может содержать повторяющиеся числа. Максимальное количество элементов в последовательности неизвестно, поэтому **последовательность необходимо поместить в вектор**. Вычислить указанные в варианте характеристики последовательности. **Написать функцию, которая решает основную задачу вашего варианта.**

Варианты задания.

1. Найти порядковый номер наибольшего по значению числа, являющегося симметричным в десятичном представлении. Пример: 1 55 31200 676 4884 30503 31412 29692 0. Результат: 30503
2. Подсчитать количество троек соседних чисел, которые можно взять в качестве длин сторон равнобедренного треугольника. Пример: 1, 8, 10, 3, 4, 4, 7, 20, 21, 21, 54, 45, 15, -18, 0. Результат: 3
3. Подсчитать количество простых чисел, которые больше своих соседних элементов справа и слева. Пример: 1 2 5 3 -5 11 8 -12 -6 -5 7 4 0. Результат: 3
4. Напечатать в обратном порядке наибольший элемент последовательности. Пример: 1 5 16 234 587 14 0. Результат: 785
5. Вычислить общую сумму всех четных цифр во всех нечетных элементах. Пример: 14 123 52 3241 5147 80 900. Результат: $2 + 2 + 4 + 4 = 12$
6. Найти наибольший элемент, имеющий в своем составе цифру 3. Пример: 134 12 5 1324 52 21345 22521 0. Результат: 21345
7. Подсчитать количество простых чисел в последовательности, больших заданного числа M. Пример: M=5, 2 3 4 7 19 18 7 8 11 15 0. Результат: 4
8. Найти сумму всех чисел, цифры в которых образуют возрастающую подпоследовательность. Пример: 9 12 32 145 231 75 512 567 112 0. Результат: $12 + 145 + 567 = 724$
9. Найти сумму всех элементов, состоящих только из разных цифр. Пример: 32 122 343 166 -2 521 0. Результат: $32 + 521 + (-2) = 551$
10. Найти количество четных чисел, сумма цифр в которых не превышает заданного числа K. Пример: K=11 345 2620 14492 1450 12206 0. Результат: 2620; 1450; 12206; количество = 3

- 11.** Найти произведение чисел, которые не превосходят 4000 и являются полными квадратами некоторого натурального числа. Пример: 2 5 4 19 25 4625 121 144 0. Результат: $4 \cdot 25 \cdot 121 \cdot 144 = 1742400$
- 12.** Найти количество элементов, которые не превосходят 10000 и делятся на одну из своих цифр (кроме цифр 0 и 1). Пример: 327 81 548 1274 43 0. Результат: (327; 548; 1274) итого:3
- 13.** Найти количество элементов, сумма цифр в которых есть простое число. Пример: 137 14 171 512 928 17 0. Результат: (137) (14) (928) количество = 3
- 14.** Найти сумму элементов, в которых чередуются четные и нечетные цифры. Пример: 2325 8122 5 1438 156 123 0. Результат: сумма $2325 + 1438 + 123 = 3886$
- 15.** Найти сумму элементов, которые состоят из двух равных частей и имеют в своем составе цифру 3. Пример: 4343 9131 33 3313 646 3223 1313 0. Результат: $4343 + 66 + 1313 = 5689$
- 16.** Найти в последовательности минимальное число, имеющее наибольшее количество цифр, отличных от 0. Пример: 4402 -2070 -2771 -3214 -4071 0. Результат: -3214
- 17.** Найти сумму таких элементов, цифры в которых не превосходят заданного числа М ($0 \leq M \leq 9$). Пример: М=7 1427 192 7 8 14 11 65 0. Результат: $1427+7+14+11+65=1524$
- 18.** Найти произведение таких элементов, которые состоят только из четных цифр (цифры 0,1,3,5,7 и 9 не являются четными цифрами). Пример: -16 -5 22 -5 14 6 15 3 -64 0. Результат: $22 \cdot 6 \cdot (-64) = -8448$
- 19.** Найти количество чисел последовательности, суммы цифр которых являются четными числами. Пример: 9 12 32 145 231 75 513 567 112 0. Результат: $5 < 145, 231, 75, 567, 112 >$
- 20.** Найти в последовательности число, имеющее наибольшее в своем составе количество четных цифр. Пример: 4402 -2070 -2771 -22814 -4071 0. Результат: -22814