

12. Стандартна бібліотека C++. Асоціативні контейнери.

Контрольні запитання:

- Які типи ітераторів існують в STL? Які з них визначені для вектору? Які до списку? Які для множини?
- Які асоціативні контейнери існують в C++? Що додає приставка multi до назви контейнера?
- Для яких стандартних класів-шаблонів не визначений метод `push_back()`? Чому? Як в ці класи додаються елементи?
- Для яких стандартних класів-шаблонів не визначений метод `push_back()`? Чому? Як в ці класи додаються елементи?
- Як визначити кількість елементів будь-якого контейнеру?
- Які коректні шляхи ітерації по вектору? Мультивідображенню? Будь-якому контейнеру?
- Як коректно пройти по всім елементам відображення?

Завдання для аудиторної роботи:

- 1) Заданий файл з текстом англійською мовою. Виділити все різні слова. Для кожного слова підрахувати частоту його входження. Слова, що відрізняються регістром літер, вважати різними. Використовувати Map.
- 2) З використанням Set виконати попарне підсумовування довільного кінцевого ряду чисел за такими правилами: на першому етапі підсумовуються попарно сусідні числа, на другому етапі підсумовуються результати першого етапу і т. д. до тих пір, поки не залишиться одне число.
- 3) У файлі записані координати точок на площині задані парою цілих чисел. Точки записані в форматі : (x1 , x2) (x1 , x2) , ... - саме так через коми та дужки. Створити файл, в якому будуть записані координати всіх відрізків з точок цього файлу, при цьому ці відрізки відсортовані за зростанням довжини.
- 4) Нехай значення функції $f(n)$ – кількість літер у письмовому представленні числа n ($f(1) = 4$ ("один"), $f(3) = 3$ ("три"), $f(42) = 8$ ("сорок два"), $f(2001) = 13$ ("дві тисячі один")). Знайдіть всі числа до 10000, для яких $f(n) = n$. (Вказівка: Використовуйте відображення (словник) для зберігання кількості літер у представленні цифри)

Завдання для самостійної роботи:

- 5) На площині задано N точок. Вивести в файл описи всіх прямих, які проходять більш ніж через одну точку із заданих. Для кожної прямої вказати, через скільки точок вона проходить. Використовувати клас `multimap`.
- 6) На площині задано N відрізків. Знайти точку перетину двох відрізків, що має мінімальну абсцису. Використовувати клас `map`.
- 7) Відредагувати задане речення, видаляючи з нього ті слова, які зустрічаються в реченні задану кількість разів.
- 8) На клітковому аркуші паперу зафарбована частина клітин. Виділити все різні фігури, які утворилися при цьому. Фігурою вважається набір зафарбованих клітин, які сусідні один з одного при руху в чотирьох напрямках. Дві фігури є різними, якщо їх не можна сумістити поворотом на кут, кратний 90 градусам, і паралельним зсувом. Використовуйте клас `multiset`.
- 9) Напишіть програму, яка знаходить усі спільні слова між двома вхідними файлами, використовуючи `set_intersection()`. Змініть його, щоб показати слова, які не є спільними, за допомогою `set_symmetric_difference()`.
- 10) З бінарного файлу зчитати вектор V_0 , ціле число $N > 0$ і набір векторів V_1, \dots, V_N . Знайти кількість векторів $V_i, i = 1, \dots, N$, в яких містяться всі елементи вектору V_0 (з урахуванням повторень). Використати алгоритм `include`.
- 11) З консолі введений рядок та вектор символів V . Знайдіть усі різні числа, які одночасно входять і в першу, і в другу половину вектору, і записати їх у текстовому файлі з заданим ім'ям у порядку заданому вектором, додаючи після кожного числа символ пробілу (використати `set_intersection`).
- 12) Даний рядок `name` і вектор V з парною кількістю цілих елементів. Знайдіть усі різні числа, які входять у другу половину вектору та при цьому відсутні в першій половині. Записати ці числа в текстовому файлі з ім'ям `name` за зростанням, виводиться кожне число на новому рядку. Використовувати алгоритм `set_difference`.
- 13) Ввести з консолі вектори рядків V_1 і V_2 . Знайти всі числа (з урахуванням повторень), які входять хоча б в один із вихідних векторів, і вивести їх в порядку незростання; при цьому, якщо, наприклад, деякий рядок входить в один із векторів 3 рази, а в інший 5 разів, то його потрібно вивести 5 разів. Використовувати алгоритм `set_union`.
- 14) Ввести з текстового файлу вектор V , що містить не менше трьох різних чисел. Виведіть усі його різні елементи, крім максимального та

мінімального, в порядку спадання за допомогою допоміжної множини та без неї.

- 15) Дан текстовий файл з ім'ям NameT. Підрахувати число повторень в ньому малих латинських літер ('a' - 'z') і створити файл з ім'ям NameS, рядки якого мають вигляд: "<літера> - < число повторень даної літери>". Літери, відсутні в тексті, в файл не включати. Рядки впорядкувати за спаданням кількості повторень літер, а при однаковій кількості повторень – по зростанню кодів літер.
- 16) Ввести вектор V, елементи якого є англійськими словами. Визначити сумарну довжину слів, що починається з однієї і тієї же літери (без урахування регістру), і вивести всі різні літери, з яких починаються елементи вектору V, разом із сумарною кількістю цих елементів (у алфавітному порядку букв); довжину виводити зразу після відповідної літери. Використовувати допоміжне відображення M, ключі якого є початковими літерами елементів вектору V, а значення - сумарна кількість цих елементів. При заповненні відображення M не використовувати умовні конструкції.
- 17) Ввести вектор цілих чисел V. Виконати групування елементів вектору V, використовуючи у якості ключу груп останню (тобто праву) цифру елементу: в одній групі повинні входити всі елементи V, що закінчуються однією цифрою (згруповані елементи повинні розташовуватися в тому же порядку, в якому вони доступні у векторі). Представити результат групування у вигляді відображення M, ключі яких є ключами групування, а значення - вектори, що містять згруповані елементи (таким чином, відображення M повинно мати тип `map <int, vector <int>`). Вивести отримане відображення (для кожного елементу відображення M спочатку вивести ключ, двокрапку, а потім елементи пов'язаного з ним вектору через кому).
- 18) Ввести вектор V, елементи якого є англійськими словами. Виконати групування елементів вектору V, використовуючи як ключ групувань другу літеру елементів (або першу - якщо слова містять лише одну літеру): В групі повинні входити всі елементи V. Представити результат групування у вигляді мультивідображення M, ключі якого є ключі групування (таким чином, відображення M повинно мати тип `multi-map <char, string>`). Вивести отримане відображення (для кожного елементу відображення M вивести ключ, а потім пов'язаний з ним елемент вектору V, ключи можуть повторюватися).
- 19) Ввести вектор дійсних чисел V. У кожній групі його елементів, що мають однакову початкову цифру, знайдіть суму значень цих елементів, за виключенням початкового елемента групи (передбачається, що елементи групи розміщуються в тому же порядку, що і у вихідному векторі). Якщо група складається з єдиного елемента, то сума повинна дорівнювати 0. Для кожної групи виводьте відповідну їй цифру і

знайдену суму, впорядкуйте пари по зростанню цифр, а потім суми.

- 20) Послідовність даних в текстовому файлі містить відомості про клієнтів клубу. Кожний елемент послідовності включає наступні дуючі цілі поля: <Код клієнту> <Рік> <Номер місяця> <Кількість занять (у годинах)> Знайти елемент послідовності з мінімальною тривалістю занять. Виведіть цю інформацію, а також відповідний рік і номер місяця (у вказаному порядку в тому ж рядку). Якщо таких елементів декілька — виведіть усі в довільному порядку.
- 21) Послідовність даних у бінарному файлі містить відомості про оцінки учнів з трьох предметів: алгебра, геометрія та інформатика. Кожен елемент послідовності містить дані про одну оцінку та має наступні поля: <Прізвище> <Ініціали> <Клас> <Назва предмета> <Оцінка>. Повних тезок (з однаковим прізвищем та ініціалами) серед учнів немає. Клас задається цілим числом, оцінка - цілим числом в діапазоні 1–12. Назва предмету вказується з заголовної літери. Для кожного учня визначити середню оцінку по кожному з предметів і вивести її з двома знаками після крапки (якщо не має жодної оцінки, то вивести для цього предмету 0.0). Відомості про кожного учня виводити на окремому рядку, вказавши прізвище, ініціали та середні оцінки по алгебрі, геометрії та інформатиці. Дані розмістити в алфавітному порядку прізвищ та ініціалів.
- 22) Реалізувати красиве виведення для всіх асоціативних контейнерів STL (map, set, multimap, multiset) за допомогою стандартного виведення, тобто потрібно перевантажити функцію `std::operator <<` для відповідних типів. Приклад виклику:

```
...
std::set<int> testSet{1,3};
std::map<int, std::string> testMap;
testMap[1] = "123";
testMap[2] = "456";
std::cout << testMap;
std::cout << testSet;
...
```

Вихідні дані (текстовий файл/виведення в консоль):

```
1=123; 2=456
1,3
```

Додаткові задачі:

- 23) Необхідно реалізувати Проху (заступник) для контейнера map, що керує доступом. Реалізувати надбудову над стандартним контейнером

map, яка дозволяла б управляти доступом до певних ключів (з розмежуванням читання, запису та видалення), а також підміняти при необхідності результати.

У цій системі є три класи:

- std :: map - базовий контейнер;
- Auditor - базовий клас «аудитора», який реалізує перевірку прав і заміну значень;
- ProxiedMap - надбудова над контейнером, яка все операції з ним передує викликом об'єкта Auditor.

Реалізувати декілька підкласів для Auditor (той що все дозволяє, що дозволяє читання та підміняє значення по заданих ключами константних значеннях і т.д.)

Розробити набір тестів, що перевіряє чи дійсно блокуються спроби несанкціонованого доступу (для різних об'єктів класу Auditor). Можливе ускладнення: Реалізувати декілька видів ітераторів (наприклад: ітератор, що обходить всі доступні ключі; ітератор, що обходить всі ключі, але сигналізує про помилку доступу при спробі звернення і т.д.)