

## 18. Стандартна бібліотека C++. Контейнери.

### Контрольні запитання:

- Створіть власний клас-шаблон `vector<T>` з методом `Норма()`. Порівняйте його дію з стандартним шаблоном `vector` в головній програмі.
- З яких частин складається бібліотека шаблонів C++?
- Для чого потрібні контейнери-адаптори? Які контейнери-адаптори визначені в C++?
- Які контейнери прямого доступу визначені в C++?
- Яка різниця між контейнерами `list`, `forward_list`, `vector`, `array`?
- Основні методи контейнеру вектор (доступ до елементів, заміна елементів, розміри)?
- Які переваги `array` або `vector` перед стандартним масивом чи вказівником?
- Як додавати елемент в вектор, стек, список?
- Як видаляти елементи в `list`, `forward_list`, `vector`, `array`?
- Які варіанти проітеруватись по елементах послідовних контейнерів?
- Як визначити кількість елементів будь-якого контейнеру?
- Які коректні шляхи ітерації по вектору? Мультівідображенню? Будь-якому контейнеру?
- Як коректно пройти по всім елементам відображення?

### Завдання для аудиторної роботи:

- 1) Біля прилавка в магазині вишикувалася черга з  $n$  покупців. Час обслуговування продавцем  $i$ -го покупця –  $t_i$ ,  $i = 1, \dots, n$ . Нехай дано натуральне  $n$  і дійсні числа  $t_1, t_2, \dots, t_n$ . Отримати  $c_1, c_2, \dots, c_n$  де з  $c_i$  – час перебування  $i$ -го покупця в черзі  $i = 1, \dots, n$ . Вказати номер покупця, для обслуговування якого продавцеві потрібно найменше часу.
- 2) Реалізувати функції для введення  $d$ -вимірних векторів ( $d$  вводиться з клавіатури). Ввести  $n$   $d$ -вимірних векторів та обчислити значення суми норм векторів.
- 3) Створіть клас-шаблон Поліном, який приймає список чисел будь-якого типу на базі стандартного класу `list` коефіцієнтів поліному.

Методи: введення-виведення, додавання, множення та обчислення значення. Перевірте, що клас працює коректно для дійсних, цілих чисел та для типу Раціональний дріб з попередніх завдань.

- 4) Ввести натуральні числа  $n, m < 100$  та матрицю розміру  $n \times m$  як список векторів (). Поміняти місцями її стовпці так, щоб їх максимальні елементи утворювали спадну послідовність.

### Завдання для самостійної роботи:

- 5) Дана матриця з цілих чисел. Знайти в ній прямокутну підматрицю, що складається з максимальної кількості однакових елементів. Використовувати клас Stack.
- 6) Реалізувати структуру «чорний ящик» на базі Queue, що зберігає множину чисел і має внутрішній лічильник K, спочатку рівний нулю. Структура повинна підтримувати операції додавання числа в множину і повернення K-го по мінімальності числа з множини.
- 7) На клітковому аркуші намальований круг. Вивести в файл опису всіх клітин, цілком лежать всередині кола в порядку зростання відстані від клітини до центру кола. Використовувати клас PriorityQueue.
- 8) На базі шаблону List реалізувати структуру зберігання чисел з підтримкою наступних операцій:
  - додавання / видалення числа;
  - пошук числа, найближчого до заданого (тобто модуль різниці мінімальний).
- 9) У вхідному файлі розташовані два набору додатних цілих чисел; між наборами – роздільник від’ємне число. Побудувати два списки C1 і C2, елементи яких містять відповідно числа 1-го і 2-го набору таким чином, щоб усередині одного списку числа були впорядковані по зростанню. Потім об’єднати списки C1 і C2 в один відсортований список.
- 10) Реалізуйте клас Auto, що містить члени: назва, модель, номер, ідентифікатор власника. Визначте для цього класу методи введення/виведення. Реалізуйте за допомогою стандартних шаблонів наступні задачі:
  - а) в шаблоні vector даний масив даних про авто, потрібно вивести всіх власників даної марки;
  - б) в шаблоні list є дані про авто, відсортуйте їх по назві та виведіть всі їх номери в цьому порядку;

- в) в шаблоні deque зберігаються дані по черги з авто на заправці — промодельуйте заповнення черги на заправці виводячи стан черги при кожному вибуванні чи прибуванні авто на заправку;
  - г) в шаблоні stack зберігаються авто на складі ринку, промодельуйте роботу складу;
  - д) використайте шаблон queue для моделювання черги з авто на мийки;
  - е) використайте шаблон priority\_queue для моделювання черги замовлень по ремонту в залежності від вартості ремонту (додатковий член класу, що вводиться окремим методом).
- 11) Складіть клас Employee із двома членами даних: hours та hourlyPay. Працівник також повинен мати функцію calcSalary(), яка повертає заробітну плату за цього працівника. Генеруйте довільну погодинну оплату праці та години для довільної кількості працівників. Зберігайте вектор Співробітник. Дізнайтеся, скільки грошей компанія витратить за даний період оплати праці.
- 12) Створіть шаблон класу Matrix, який створений з вектору <vector<T>>. Надайте його дружньому методу ostream & operator << (ostream &, const Matrix &) для відображення матриці. Створіть наступні бінарні операції, використовуючи об'єкти функції STL, де це можливо: оператор + (const Matrix &, const Matrix &) для додавання матриці, оператор \* (const Matrix &, const vector<int> &) для множення матриці на вектор та оператор \* (const Matrix &, const Matrix &) для множення матриць. Перевірте шаблон класу Matrix, використовуючи int і float.
- 13) Реалізувати функцію, що виконує додавання чисел, заданих вектором unsigned char в різних системах числення:
- ```
vector<int> addition(const vector<UCHAR> & A, int baseA, const vector<UCHAR> & B, int baseB, int baseResult);
```
- Функція повинна перевіряти вхідні дані про коректність і повертати пустий вектор у разі виявлення помилки. В текстових файлах записані перше та друге число та останнім числом — основа числення. Основа числення результату вводиться з консолі та результат записується в третій файл. Тестування
- Для основних функцій плюс повинен бути створений набір тестів, що перевіряють функції на наборі прикладів, та коректність введення в разі некоректних даних.
- Додатково реалізуйте також підтримку записів вхідних даних у рядках. Наприклад
- ```
16: "FF"
```

10: "256"

2

Результат: "111111111"

- 14) Даний текстовий файл, що містить рядкові представлення цілих чисел. Заповнити вектор  $V$  числами з цього файлу та вивести їх у вихідному порядку. У випадку некоректних даних видайте змістове повідомлення.
- 15) В консолі вводиться масив цілих чисел. Заповнить список  $L$  цими числами і вивести елементи списку  $L$  в початковому порядку у вихідному, а потім в оберненому порядку. Відсортуйте дані за зростанням у списку, але виведіть навпаки за спаданням.
- 16) Даний вектор цілих чисел з парною кількістю елементів. Заповнить дек  $D$  даними числами так, щоб перша половина чисел збігалася з порядком заповнення вектору, а друга була в зворотному порядку.
- 17) Ввести список цілих чисел з консолі. Вставити перед кожним ненульовим елементом вихідного списку число  $-1$ , а після кожного рівного  $2$  – нуль.
- 18) Ввести з консолі список  $L$  натуральних чисел. Вставити після кожного непарного елемента з першої половини вихідного списку число  $-1$ , а перед кожним парним елементом другої половини  $-2$ .
- 19) Ввести з текстового файлу дек довільного типу  $D$ . Видалить середній елемент дека, якщо кількість елементів непарна або 2 середні елементи – якщо парна.
- 20) Ввести з текстового файлу дійсний вектор  $V$  з непарною кількістю елементів  $N$  ( $N \geq 5$ ). Якщо там парна кількість елементів – додати до вектору 3 дійсні числа з консолі. Видалити три середні елементи вектора за один виклик `erase`.
- 21) Ввести список  $L$  з консолі та вектор  $V$  з бінарного файлу (тип – рядок). Перемістити елемент списку  $L$  з даним номером в кінець списку  $V$ .
- 22) Ввести список  $L$  з елементами  $A_1, A_2, \dots, A_{N-1}, A_N$  ( $N$  – парне, якщо ні, то додайте нуль до списку). Змінити порядок елементів у списку на наступний:  $A_1, A_N, A_2, A_{N-1}, A_3, A_{N-2}, \dots, A_{N/2}, A_{N/2-1}$ .
- 23) Ввести два списки  $L1$  і  $L2$  з однаковою кількістю елементів –  $N$ . Якщо це не так, то видалить з кінця більшого списку потрібну кількість елементів. Отримати в списку  $L2$  комбінований набір елементів – список вигляду  $B_1, A_1, B_2, A_2, \dots, B_N, A_N$ , де  $A_I$  – елементи вихідного списку  $L1$ , а  $B_I$  – елементи списку  $L2$ . Вказівка. Використайте `splice` для  $L2$  з інкрементами у другому та третьому аргументах.