**Лабораторна робота №8.** **Написання програм з використанням рядкових величин та структур**

**На період** карантину в дистанційній формі навчання потрібно використовувати замість **С++ Builder онлайн компілятор C++ Shell, який доступний за адресою** [**http://cpp.sh**](http://cpp.sh) (інструкція надана після тексту ЛР№3). При наявності на домашньому комп’ютері іншого компілятора завдання можна виконати на ньому. Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді URL-посилання для робіт, виконаних у **C++ Shell,** або для робіт, на компіляторі С++ домашнього комп’ютера cpp-файл з іменем у форматі

**АР<Номер групи><Номер практичної / лабораторної [літера позначення типу роботи P – практична, R – лабораторна]<Прізвище англійською>**. Наприклад, **АРPTBD-2107P**buts.срр.

При відсутності можливості використовувати компілятор або відсутності можливості доступу до Інтернету текст програми набрати в Блокноті або WordPad Windows, в крайньому випадку текст програми можна записати у зошити, сфотографувати, оформити відповідним чином (включити титульний лист, записати завдання, а потім включити файли з фотографіями) та надіслати на електронну адресу викладача.

Оформлені роботи повинні надсилатися у вигляді текстового документу з іменем файлу:

**АР<Номер групи><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**. Наприклад, **АРPTBD-2104P**buts.doc.

**Тему в заголовку листа записати**

**АР <Номер групи>-><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**

**Строк відсилки ЛР 07.04.2021.**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, тему в заголовку листа записати

**АР<Номер групи>-Запитання-<Прізвище англійською>**.

**Мета**: навчитися писати програми з використанням символьних та рядкових величин, структур та масивів **на мові С++** в **консольному режимі.**

**Методичні вказівки щодо організації самостійної роботи студентів**

1. Запустити середовище програмування С++ .
2. Записати програму, що виконує 3 завдання з пп.3.1-3.3. В першому рядку кожної програми записати

*// ПТБД-21 Група № Прізвище*

вказавши номер своєї групи та своє прізвище.

Як і в ЛР №5, програма повинна запитати номер завдання (число 1, 2, 3 або 0 для закінчення) і в залежності від введеного значення виконувати відповідне завдання. Якщо введений 0 – програма припиняє роботу.

Вхідні дані ввести, а результати вивести, використовуючи потокове введення-виведення даних.

1. Завдання:
   1. Написати код, який заповнює масив з 10 елементів випадковими числами з діапазону від 200 до 300, виводить всі елементи масиву з вказівкою індексу масиву, розділивши їх табуляцією, виконує пошук мінімального та максимального числа та їх виведення.
   2. Написати код, який заповнює масив з 10 елементів випадковими числами з діапазону від -20 до 20, виводить всі елементи масиву з вказівкою індексу масиву, розділивши їх табуляцією, виконує заміну всіх негативних чисел на 55, а позитивні робить негативними. Вивести отриманий масив аналогічно заповненому.
   3. Написати код, який заповнює по 2 змінні (можна ввести з консолі або проініціалізувати) типу структура з інформацією про особу. Структура повинна містити поля:

* код (порядковий номер),
* прізвище та ініціали,
* стать (1 – жіноча, 2 чоловіча),
* телефон,
* адреса,
* професія.

**Теоретичні відомості (додаткові пояснення до раніш наданого матеріалу).**

Масив являє собою сукупність даних, що організована певним чином. Тобто масив – це окремий випадок структури даних.

**Властивості масивів**

* масив складається з елементів, які мають однаковий тип;
* елементи масиву послідовно, без проміжків, розташовані в одній ділянці оперативної пам’яті;
* кожен з елементів масиву має свій порядковий номер, що зветься індексом;
* нумерація елементів починається з 0;
* до елементів масиву можна звертатися використовуючи ім’я масиву і індекс у квадратних дужках.
* масив може бути одновимірним, або багатовимірним,
* У двовимірному масиві кожний з елементів є одновимірним масивом;
* У тривимірному масиві кожний з елементів є двовимірним масивом.

Оголошення одновимірного масиву

**<тип елементів> <ім’я масиву> [<кількість елементів>] ;**

Приклади оголошень масивів:

int arr [10] ;

int A [10] ={2,5,10};

int A [] ={2, 5, 10, 3, 6, 0, 9, 4, 5, 7};

Доступ до елементів масиву

Для доступу до елементів масиву використовується синтаксична конструкція, що складається з імені масиву та індексу, який записується у квадратних дужках.

Наприклад, A[i], A [0], A [9].

Індексовані елементи масиву можуть бути використані так само, як і прості змінні. Вони можуть бути операндами у виразах, їм можна присвоювати значення, відповідні їх типу.

int arr[ ] ={2, 5, 10, 3, 6, 0, 9, 4, 5, 7}



Масиви символів

* У мові С++ символьний рядок може розглядатися як масив символів.
* Для масиву символів у пам’яті виділяється ділянка, розмір якої на один байт більше, ніж кількість символів у рядку.
* У цей додатковий байт записується ознака кінця рядка - символ ‘\0’.

char <ім’я символьного рядка> [<кількість байтів >] ;

Приклад:

char str[80] ;

char str[10] ={‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘!’, ‘\0’} ;

char str[10] =”Hello!”;

char str[] =”Hello!”;

Особливості роботи з масивами в С++

* Ніякого контролю за значеннями індексів, що використовуються для доступу до елементів масиву, нема.
* Ви можете звернутися до «елементу масиву» з номером, який більше максимального, але отримаєте невідомо що. Ще гірше, якщо ви щось напишете за цим номером

Приклади

**Заповнення масиву випадковими числами**

В С ++ для цього є спеціальні фyнкції **rand()**. Вони знаходяться в бібліoтечному файлі cstdlib, тому щоб їх застосовувати в програмі, необхідно підключити цей бібліотечний файл: #include <cstdlib>.

Окрім того, для ця функція може використовувати поточний системний час через використання функції time(), для якої необхідно підключити бібліотечний файл ctime: #include <ctime>.

Діапазон випадкових чисел, які генеруються таким чином – від 0 до 32767. Можливо вам знадобиться заповнити масив числами від 200 дo 300, від 0.1 дo 1, від -20 дo 20. Таку генерацію випадкових чисел можливо і нескладно реалізувати. У прикладі нижче розглянуто кілька випадків:

**#include <iostream>**

**#include <cstdlib>**

**#include <ctime>**

**using namespace std;**

**int main()**

**{**

**int randomDigits[10];**

**int randomDigits\_2[10];**

**int randomDigits\_3[10];**

**int randomDigits\_4[10];**

**float randomDigits\_5[10]; // для чисел з плаваючею точкою**

**srand(time(NULL));**

**for (int i = 0; i < 10; i++)**

**{**

**randomDigits[i] = rand() % 7; // 0 ... 6**

**randomDigits\_2[i] = 1 + rand() % 7; // 1 ... 7**

**randomDigits\_3[i] = 200 + rand() % 101; // 200 ... 300**

**randomDigits\_4[i] = rand() % 41 - 20; // -20 ... 20**

**randomDigits\_5[i] = 0.01 \* (rand() % 101);// 0.01 ... 1**

**}**

**cout << "Масив з числами від 0 до 6: ";**

**for (int i = 0; i < 10; i++)**

**{**

**cout << randomDigits[i] << " ";**

**}**

**cout << endl << "Масив з числами від 1 до 7: ";**

**for (int i = 0; i < 10; i++)**

**{**

**cout << randomDigits\_2[i] << " ";**

**}**

**cout << endl << "Масив з числами від 200 дo 300: ";**

**for (int i = 0; i < 10; i++)**

**{**

**cout << randomDigits\_3[i] << " ";**

**}**

**cout << endl << "Масив з числами від -20 до 20: ";**

**for (int i = 0; i < 10; i++)**

**{ cout << randomDigits\_4[i] << " "; }**

**cout << endl << "Масив з числами від 0.01 дo 1: ";**

**for (int i = 0; i < 10; i++)**

**{ cout << randomDigits\_5[i] << " "; }**

**cout << endl;**

**}**

**Виведення масиву на консоль**

**int const size=10;**

**int ar[size];**

**/\* Здійснити заповнення масиву через введення даних або ініціалізацію \*/**

**for(int i=0; i<size; i++){**

**cout<<ar[i];**

**if(i<size-1) cout<<'\t';**

**}**

**cout<<endl;**

**Обчислення суми елементів масиву**

**int const size =10;**

**int ar[size];**

**/\* Здійснити заповнення масиву через введення даних або ініціалізацію \*/**

**int sum=0;**

**for (int i = 0; i < size; ++i) {**

**sum+=ar[ i ];**

**}**

**cout<< "sum="<< sum << endl;**

**Пошук найбільшого елементу у масиві**

**int const size =10;**

**int ar[size];**

**/\* Здійснити заповнення масиву через введення даних або ініціалізацію \*/**

**int mx=0;**

**for(int i=0; i<size;i++)**

**if (ar[i]>mx) mx=ar[i];**

**cout<< "mx="<< mx<<endl;**

**Пошук найменшого елементу у масиві та його індексу**

**int const size =10;**

**int ar[size];**

**/\* Здійснити заповнення масиву через введення даних або ініціалізацію \*/**

**int mn=ar[0];**

**for(int i=1; i<size;i++)**

**if (ar[i]<mn) mn=ar[i];**

**cout<< "mn="<< mn<<endl;**

**Вилучення елементу із масиву**

**int const size =10;**

**int ar[size];**

**for(int i=0; i<size; i++) {ar[i]=200+ rand() % 5;**

**cout<<ar[i]<<endl;**

**}**

**int element = 200;**

**int size1 = size;**

**/\* Здійснити заповнення масиву через введення даних або ініціалізацію \*/**

**int i=0;**

**while(i < size1){**

**if(ar[i] == element) {**

**size1 = size1 - 1;**

**for(int j = i; j<size1; j++)**

**ar[ j ] = ar[ j+1 ];**

**}**

**else i++;**

**}**

**Вставка елементу до масиву**

**int x = 55;**

**int pos = 3;**

**int size1 = size;**

**/\* Здійснити заповнення масиву через введення даних або ініціалізацію \*/**

**if(pos<0 || pos>size) cout << "Bad position "<<endl;**

**else {**

**for(int i=size1;i>pos;i--)**

**{ ar[i]=ar[i-1];**

**ar[pos]=x;**

**size1++; }**

**};**

**Операції над рядками символів**

* Пошук позиції символу у рядку;
* пошук позиції під рядка у рядку;
* копіювання частини рядка;
* вилучення частини рядка;
* вставка підрядка у рядок;
* перетворення числа у рядок символів;
* перетворення рядка символів у число.

**Пошук позиції символу у рядку**

**char str[] = " Bad news has wings ";**

**char ch='s';**

**int NumPos=-1; //Номер позиції шуканого символу;**

**// =-1 символ не знайдений**

**for(uint i=0; str[i] != '\0'; i++)**

**if(str[i]==ch) { NumPos= i; break;}**

**cout << "Position "<< NumPos <<endl;**

***Можливі труднощі в лабораторній роботі №8***

В ***лабораторній роботі №8*** два завдання. Ви могли їх виконувати за тією ж схемою, що і в попередніх роботах – через оператор вибору switch. При використанні оператору вибору switch часто "загублюються" фігурні дужки, і з’являються помилки компіляції, що викликає розгубленість, тому важно перевіяйте щоб кожній дужці { відповідала дужка }.

Якщо уважно подивитися, вони вирішуються майже за однаковим алгоритмом. Розглянемо як можна виконати ці 2 завдання в оптимальний спосіб.

Не забуваємо про включення бібліотек та using namespace std;

Що робимо в main():

Команда srand(time(NULL));

Оголошуємо змінну для вибору режиму.

Оголошення масиву з 10 елементів (див. приклад в ЛР№8 на стор.2). Ви для обох завдань можете використовувати один масив, оскільки після виконання режиму програма закінчує роботу, цей значення з цього масиву більше не потрібні.

Через оператор вибору switch обираємо режим.

Виконуємо дії відповідно до режиму.

Виконує або cin.get(); або system ("pause"); для виконання затримки екрану, *якщо компілятор цього не робить*.

Якщо у вас було оголошено int main(), то виконуєте return 0;

Що робимо в операторі вибору case.

1. ***Завдання 3.1***
2. Оголошення змінних. Для визначення мінімального та максимального значення вам потрібно дві додаткові змінні. На самому початку ви вважаєте, що перший елемент (а в масиві це елемент з індексом 0) є мінімальний для пошуку мінімального, та максимальний для пошуку максимальний. Поки ви не порівняєте з наступним, ви не знаєте який він.
3. Виконання підготовчих дій

* Виводимо заголовок для переліку заповнених значень масиву
* В циклі заповнюємо та одразу виводимо значення.

1. Виконання аналізу та коригування значень масиву.

* В циклі проводимо аналіз кожного поточного значення, починаючи з 1 (нульовий елемент ми зафіксували як початкове значення).
* Якщо поточне значення більше максимального, то вважаємо його максимальним (присвоюємо це значення обраній змінній для збереження максимального значення), *інакше* – перевіряємо чи є поточний елемент мінімальним, тобто меншим за значення відповідної змінної. Якщо він менший, то присвоюємо це значення обраній змінній для збереження мінімального значення.

1. Виведення результату – мінімальне та максимальне значення.
2. ***Завдання 3.2***
3. Виконання підготовчих дій - аналогічні дії як для завдання 3.1 та окрім цього виводимо заголовок для зміненого масиву.
4. Виконання аналізу та коригування значень масиву та виведення результату.

* Якщо поточне значення позитивне, змінюємо його на негативне (пам’ятаємо, що для int a=1; a=-a; отримаємо негативне a=-1?),  *інакше* – перевіряємо чи є поточний елемент позитивним, і якщо -так, замінюємо його на 55.
* Після цих дій виводимо значення поточного елементу масиву.

1. ***Завдання 3.3***

Вам потрібно створити структуру. Погляньте на приклад в лекції і створіть за аналогією.

*Контрольні запитання для самоперевірки*.

1. Що таке масив та які існують різновиди масивів?
2. Як здійснюється звернення до елементів масивів?
3. Як у С++ реалізується введення-виведення елементів масиву?