**Лабораторні роботи 08-09. Проблемні питання.**

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.

В **лабораторній роботі №9** вам потрібно створити структуру. Погляньте на приклад 1 в теоретичних відомостях і створіть за аналогією.

Які у вас будуть поля? Місто та вулиця змінні типу string, номер будинку, кількість квартир – ціле, інтернет - логічна/булева змінна (тип bool, значення true або false, а ще які пам’ятаєте? Якщо ні то Лекція 7, починаючи зі стор.5).

Як оголосити масив структур – див. приклад 3 з теоретичних відомостей. Там оголошується **Point M[10];** Ви замість **Point** вказуєте ім’я своєї структури, відповідно можете дати своє ім’я масиву.

Працюєте з масивом структур як у на приклад 2 в теоретичних відомостях. Дані кожному елементу структури ви можете присвоювати безпосередньо, а можна через список ініціалізації.

Розглянемо цей момент стосовно прикладу 1. Якщо нам потрібно створити масив книжок з 3-х елементів та вивести всі дані, то це можна зробити так:

**// Example program**

**#include <iostream>**

**#include <string>**

**#include <cstdlib>**

**#include <ctime>**

**using namespace std;**

**int main()**

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

**struct Book**

**{**

**string title;**

**string author;**

**int year;**

**float price;**

**};**

**// Заповнення масиву структур**

**Book PrgBook[3] = {"1111", "aaaa", 200 + rand() % 101, 10.0,**

**"2222", "bbbb", 200 + rand() % 101, 20.0,**

**"3333", "cccc", 200 + rand() % 101, 30.0,};**

**// Виведення масиву**

**for (int i=0; i < 3; i++) cout << PrgBook[i].title << '\t' << PrgBook[i].author << endl;**

**return 0;**

**}**

Ви можете зробити аналогічне заповнення масиву структур, а може заповнювати таким чином :

**PrgBook[0].title="Вступ до програмування мовою С++";**

**PrgBook[0].author="Ю. А. Бєлов";**

**PrgBook[0].year=2012;**

**PrgBook[0].price=45.50;**

Решта елементів заповнюється аналогічно.

Після заповнення значень масиву структур з описом будинку. Ви запитуєте скільки значень вивести з існуючих (вказуєте розмір свого масиву структур), і якщо більше то самі вирішуйте або припиняйте роботу з повідомленням щодо неправильної кількості, або давайте повідомлення про існуючу кількість і вважайте її за введену. Виводите значення елементів масиву структур відповідно до заданої кількості. Пам’ятаємо, що індексація масивів починається з нуля.

Після цього спочатку визначаємо лічильник кількості будівель без інтернету з початковим значенням 0, виводимо заголовок "Будівлі з інтернетом", а потім в циклі аналізуємо всі елементи масиву структур на предмет наявності інтернету. Якщо в поточному елемент стоїть ознака відсутності інтернету підраховуємо кількість будівель без інтернету збільшуючи на одиницю лічильник, якщо інтернет є - виводимо порядковий номер в списку (тобто індекс циклу збільшений на 1, пам’ятаємо про індексацію масивів!), місто, вулицю, номер будинку.

Якщо значення лічильника кількості будівель без інтернету відповідає кількості заповнених вами елементів масиву структур, то виводите текст "Відсутні". Звертаю увагу, якщо ви оголосили масив з певною кількістю елементів, але заповнили не все, то ви повинні зафіксувати заповнену кількість і саме вона у вас буде верхньою межею в перегляді елементів масиву.

Після цього виводите текст "Кількість будівель без інтернету" та їх кількість.