Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»

Звіт до лабораторної роботи № 17

З дисципліни «Програмування частина 2»

Дослідження особливостей використання вказівників у мові С

Виконав студент групи АП-12:

Дроф’як Святослав

Перевірив викладач:

Пелех Н.В.

Львів 2024

Мета роботи: ознайомитися з поняттям вказівник та особливостями його використання у процесі програмування.

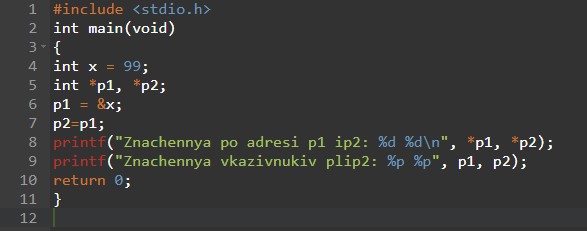
# Хід роботи

1. Ознайомитися з теоретичними відомостями.
2. Здійснити виконання прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати виконання у звіті.
3. Написати программу для визначення суми масиву, який складається з п’яти елементів. Значення елементів вводяться з клавіатури. Для доступу до елементів масиву використати вказівник. Представити скріни коду програми та результати її виконання у звіті.
4. Написати функцію, яка визначає кількість додатних елементів масиву цілих чисел. Параметрами функції є масив і його розмірність. Використовуючи написану функцію, обчислити і вивеси на екран кількість додатних елементів масиву цілих чисел розмірністю 13. Елементи масиву ввести з клавіатури.
5. Оформити звіт.

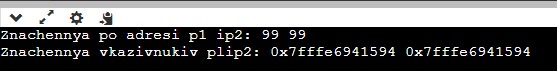
**Завдання 1**

**Реалізація:**

Програмна реалізація мовою С



Код завдання 1, приклад 1

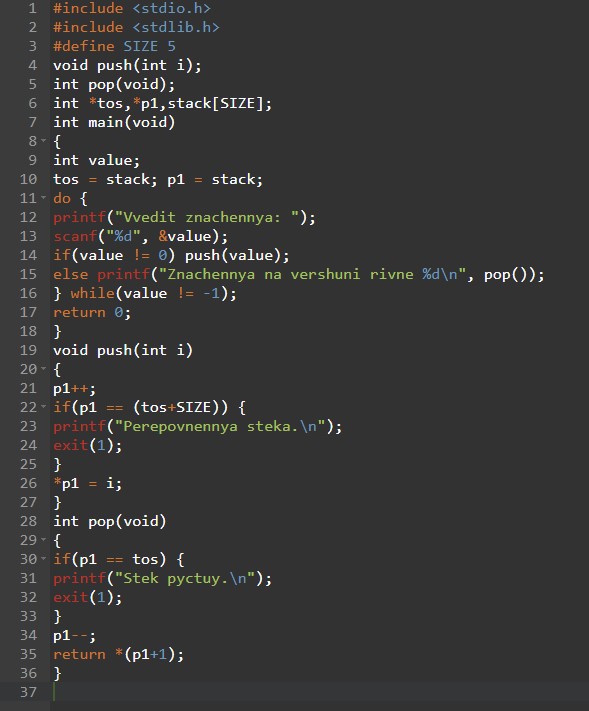


Скріншот виконання коду, приклад 1

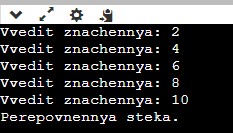
Ця програма демонструє використання **вказівників** та **адрес пам’яті** в мові **C**. Ось коротке пояснення:

1. Спочатку оголошується ціле число x і присвоюється значення 99.
2. Далі оголошуються два вказівники на цілі числа: p1 і p2.
3. Вказівник p1 отримує адресу пам’яті змінної x (тобто він вказує на x).
4. Вказівник p2 отримує значення вказівника p1 (тобто він також вказує на x).
5. Програма виводить значення, на яке вказують вказівники p1 і p2 (обидва значення - 99).

Також виводяться адреси пам’яті вказівників p1 і p2.



Код завдання 1, приклад 2

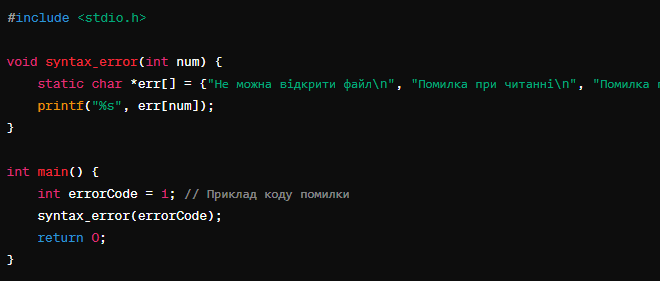


Скріншот виконання коду, приклад 2

Ця програма реалізує **стек** (або “стопку”) за допомогою вказівників та масиву в мові **C**. Ось коротке пояснення:

1. Спочатку оголошується масив цілих чисел під назвою stack з фіксованим розміром 5.
2. Два вказівники оголошуються: tos (вершина стеку) та p1 (використовується для додавання та видалення елементів).
3. Вказівники tos та p1 спочатку вказують на початок масиву stack.
4. Програма входить у цикл, де користувач може вводити значення:
   * Якщо введене значення не дорівнює 0, воно додається до стеку за допомогою функції push.
   * Якщо введене значення дорівнює 0, програма виводить значення з вершини стеку за допомогою функції pop.
   * Якщо введене значення дорівнює -1, цикл завершується.
5. Функція push збільшує вказівник p1 та додає введене значення до стеку.
   * Якщо стек заповнений, виводиться повідомлення про переповнення та програма завершується.
6. Функція pop зменшує вказівник p1 та повертає значення з вершини стеку.

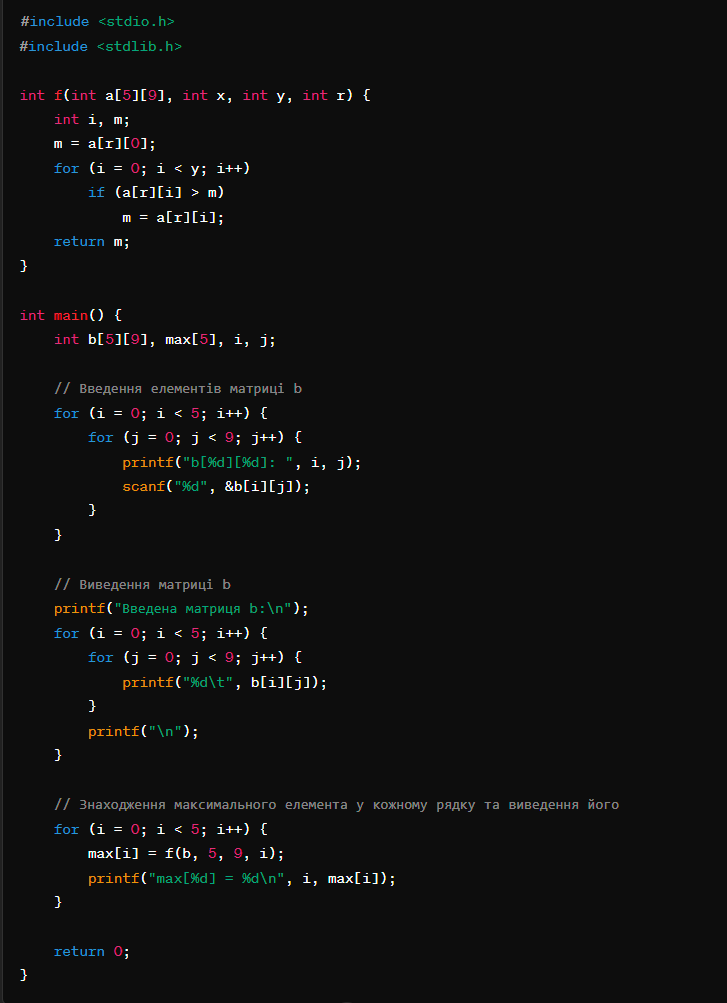
Якщо стек порожній, виводиться повідомлення про викидання та програма завершується.



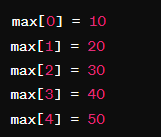
Код завдання 1, приклад 3

****

Скріншот виконання коду, приклад 3

****

Код завдання 1, приклад 4

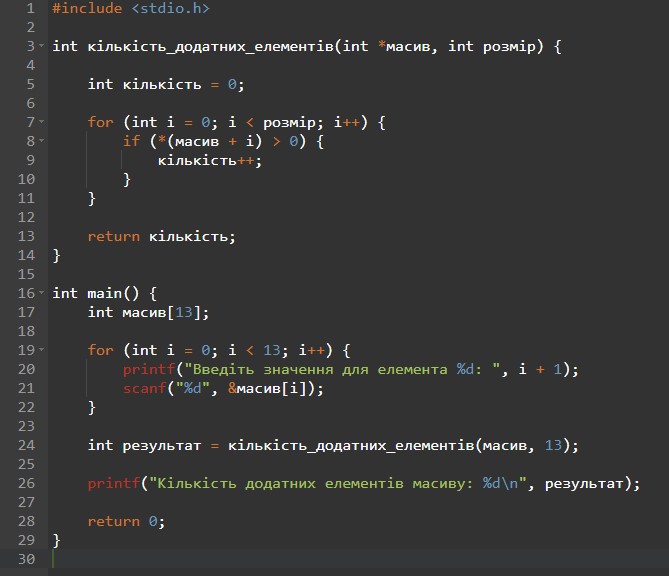
****

Скріншот виконання коду, приклад 4

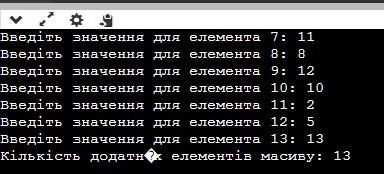
**Завдання 2**

**Реалізація:**

Програмна реалізація мовою С



Код завдання 1, приклад 2



Скріншот виконання коду, приклад 2

Ця програма реалізує обчислення **кількості додатних елементів масиву** в мові **C**. Ось коротке пояснення:

1. Спочатку оголошується масив цілих чисел під назвою масив з розміром 13.
2. Користувач вводить значення для кожного елемента масиву.
3. Функція кількість\_додатних\_елементів приймає масив та його розмір.
4. У циклі перевіряється, чи є кожен елемент додатним числом.
5. Якщо так, збільшується лічильник кількості додатних елементів.

Результат кількості додатних елементів виводиться на екран.

# Контрольні питання

1. Дайте визначення поняття вказівник.

**Вказівник** - це змінна, яка містить **адресу пам’яті** іншої змінної. Вказівники дозволяють працювати з пам’яттю напряму, звертаючись до значень за їхніми адресами.

1. Які арифметичні операції можуть виконуватись з вказівниками?

**Арифметичні операції з вказівниками**:

* **Додавання**: Вказівники можна додавати або віднімати, щоб переміщатися по пам’яті.
* **Збільшення/зменшення**: Збільшення вказівника на 1 вказує на наступний елемент масиву (або наступний байт пам’яті).
* **Віднімання вказівників**: Віднімання вказівників допомагає обчислити розмір між двома адресами.

1. Поясніть призначення функції pop () у стеку.

**Функція**pop()**у стеку**:

* Функція pop() використовується для видалення елемента з вершини стеку.
* Вона зменшує вказівник вершини стеку, щоб вказувати на попередній елемент.
* Повертає значення, яке було видалено зі стеку.

1. Які є методи звертання до елементу масиву?

**Методи звертання до елементу масиву**:

* **За індексом**: Використовується квадратні дужки, наприклад, масив[індекс].
* **За вказівником**: Вказівник може вказувати на елемент масиву, наприклад, \*(вказівник + індекс).

1. Які переваги використання прийомів адресної арифметики при зверненні до елементів масиву?

**Переваги використання прийомів адресної арифметики при зверненні до елементів масиву**:

* **Швидкість**: Адресна арифметика дозволяє ефективно обробляти масиви без зайвих обчислень.
* **Ефективність пам’яті**: Вказівники дозволяють зберігати адреси, а не копії значень, що економить пам’ять.
* **Гнучкість**: Вказівники дозволяють динамічно обробляти дані, змінюючи адреси.

Висновок:

У цій лабораторній роботі я отримав можливість ознайомитися з важливим поняттям вказівника в мові програмування C та його застосуванням у практичних завданнях. Для мене важливо зрозуміти, що вказівники є потужним інструментом у C, який дозволяє працювати з пам'яттю низького рівня та отримувати більшу ефективність та гнучкість у розробці програм.

Під час виконання лабораторної роботи я спочатку ознайомився з теоретичними аспектами вказівників: їх призначенням, синтаксисом та основними правилами використання. Далі я виконав декілька прикладів, щоб закріпити свої знання, та перейшов до практичних завдань.

У другій частині я написав програму для обчислення суми масиву, використовуючи вказівники. Це дозволило мені застосувати теоретичні знання у практичній ситуації та зрозуміти, як працювати з вказівниками у реальних завданнях програмування.

Початок форми