

ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА

Факультет прикладної математики та інформаційних технологій

Кафедра обчислювальної математики та математичної кібернетики

**Лабораторна робота №1**

з дисципліни «Алгоритми та структури даних»

на тему «Лінійні списки: створення, перегляд»

Варіант №4

Виконала:

студентка групи ПС-24-1

Анна Гарт

Дніпро, 2025

### ***Відповіді на контрольні запитання:***

1. Список – це лінійна структура даних, що складається з набору елементів (вузлів), які можуть бути розташовані в довільному порядку в пам'яті, але логічно зв'язані між собою за допомогою посилань (вказівників). Списки бувають однозв'язними, двозв'язними та кільцевими.

2. Не кожен список є зв'язним. Існують також масиви, які також є списками в загальному розумінні (набором елементів одного типу), але вони зберігаються в послідовних комірках пам'яті та не містять явних посилань між елементами. Зв'язний список – це конкретна реалізація списку з використанням вказівників.

3. Перший елемент (голова списку) має таку відмінність: на нього посилається зовнішній вказівник head, тоді як наступні елементи отримують посилання від попередніх через поле next. Також, якщо список порожній, head = NULL.

4. Останній елемент має поле next, яке містить значення NULL, що сигналізує про кінець списку. Усі інші елементи мають у next адресу наступного вузла.

5. Оскільки доступ до елементів списку можливий лише послідовно, починаючи з першого елемента, позиціонування на нього забезпечує можливість обходу всього списку. Без збереження посилання на перший елемент список буде втрачено.

6. Перевірка if (head == NULL) необхідна для уникнення помилок доступу до пам'яті (наприклад, спроби читання чи видалення елементів з порожнього списку). Вона також допомагає коректно виконувати операції додавання першого елемента.

**Мета виконання завдання:** навчитися створювати та опрацьовувати лінійні списки, реалізовувати їх програмно та виконувати базові операції над елементами списку (додавання, перегляд, підрахунок характеристик).

**Постановка завдання.** Варіант 4. Розробити програму зі створення й опрацювання зв'язних лінійних списків. Вводити числа до списку, доки не введено число 999. Обчислити процентний вміст додатних, від'ємних та нульових елементів утвореного списку. Зобразити створені списки у протоколі лабораторної роботи.

### ***Текст програми:***

```
/* Гарт Анна Вікторівна, група ПС-24-1, варіант №4
 * Лабораторна робота №1. Лінійні списки: створення, перегляд */

#include <iostream>
#include <stdio.h> // підмикання бібліотеки stdio.h; у програмі використано функцію printf()
#include <stdlib.h> // підмикання бібліотеки stdlib.h; у програмі використано функцію system()
using namespace std;

struct Node {
    int data;          // поле для збереження значення (цілого числа)
    Node *next;        // покажчик на наступний елемент списку
};

int main() {
    Node* head = NULL; // Вказівник на початок списку
```

```

Node* c;           // Поточний елемент списку
Node* end = NULL;  // Вказівник на останній елемент списку

int countPositive = 0; // лічильник додатних чисел
int countNegative = 0; // лічильник від'ємних чисел
int countZero = 0;     // лічильник нулів
int totalCount = 0;    // загальна кількість елементів

system("chcp 65001");
system("cls");

cout << "Введіть список чисел (для завершення введіть: \"999\" і натисніть
Enter):" << endl;

// Цикл введення та створення списку
do {
    c = new struct Node; // Створення нового елемента
    c->next = NULL;      // Поки що не має наступного
    cin >> c->data;       // Зчитування числа

    // Підрахунок статистики перед додаванням у список
    if (c->data > 0) {
        countPositive++;
    } else if (c->data < 0) {
        countNegative++;
    } else {
        countZero++;
    }
    totalCount++;

    // Додавання елемента до списку
    if (head == NULL) {
        head = c;
        end = c;
    } else {
        end->next = c;
        end = c;
    }
} while (c->data != 999); // Доки не введено -1000

// Виведення списку на екран
cout << "\nСписок чисел:" << endl;
c = head;
while (c != NULL) {
    cout << c->data << " ";
    c = c->next;
}
cout << endl;

// Обчислення процентів
double percentPositive = 0, percentNegative = 0, percentZero = 0;

```

```

if (totalCount > 0) {
    percentPositive = (double) countPositive / totalCount * 100;
    percentNegative = (double) countNegative / totalCount * 100;
    percentZero = (double) countZero / totalCount * 100;
}

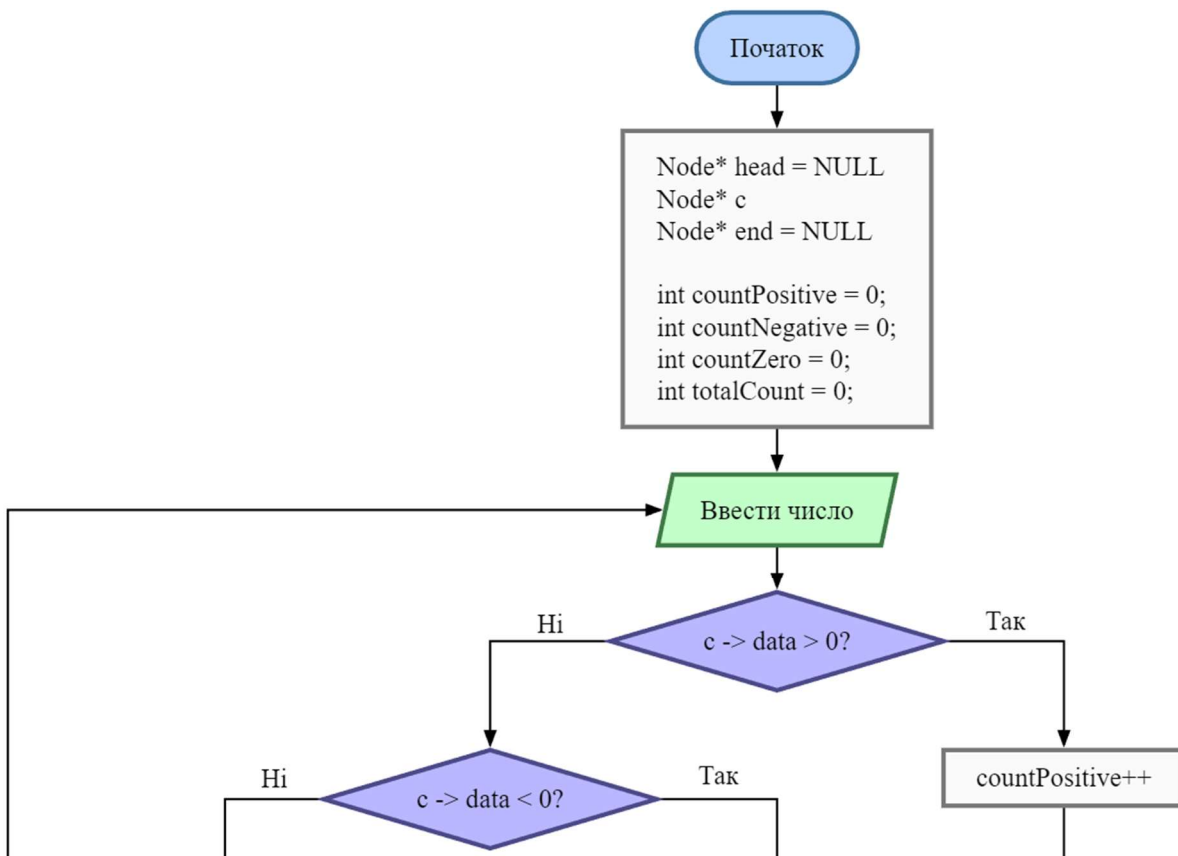
// Виведення статистики
printf("\n=== Статистика ===");
printf("\nЗагальна кількість елементів: %d", totalCount);
printf("\nДодатних чисел: %d (%3.2f%)", countPositive, percentPositive);
printf("\nВід'ємних чисел: %d (%3.2f%)", countNegative, percentNegative);
printf("\nНулів: %d (%3.2f%)", countZero, percentZero);

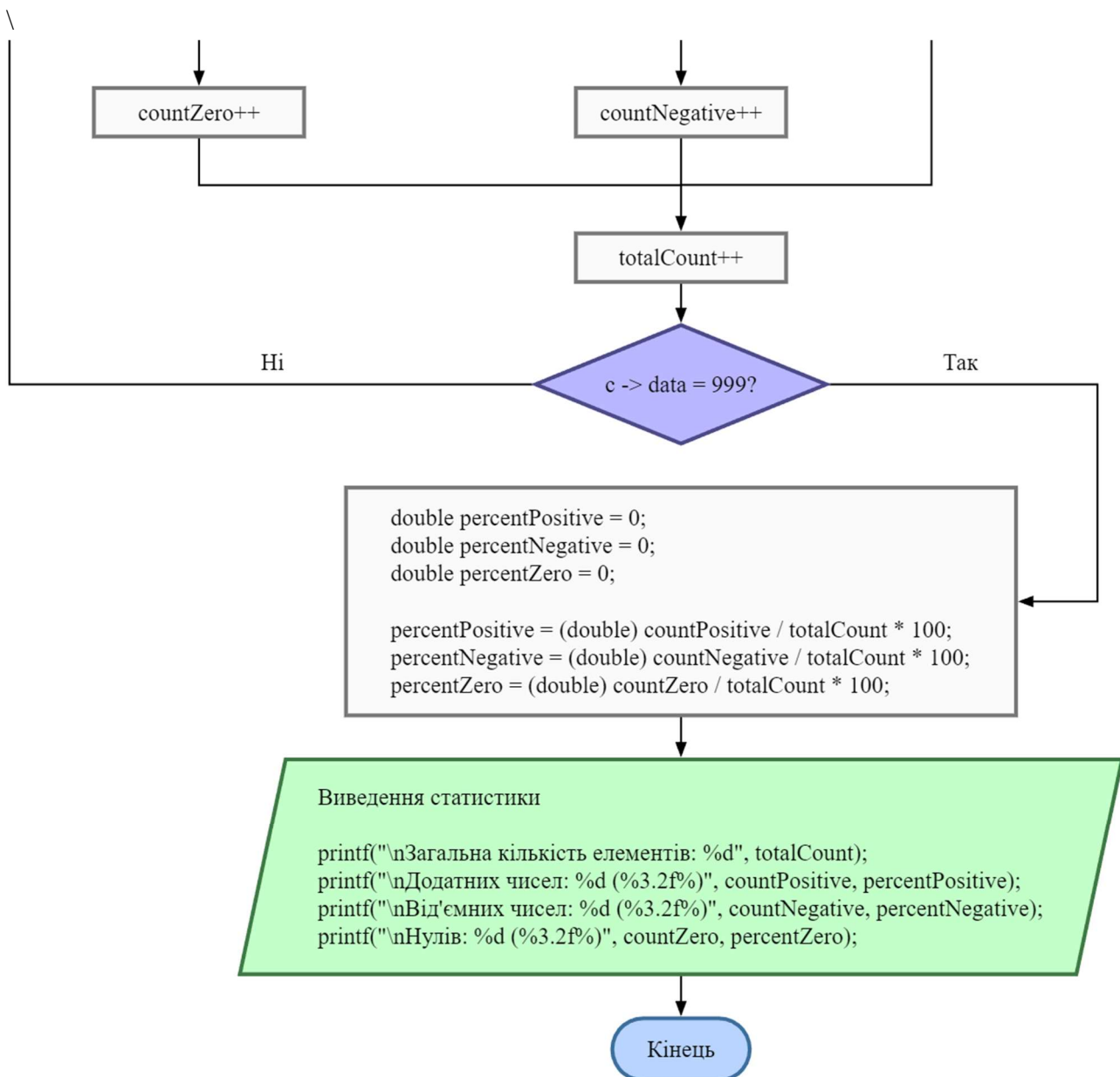
// Очищення пам'яті
c = head;
while (c != NULL) {
    Node* temp = c;
    c = c->next;
    delete temp;
}

printf ("\n\n");
system("pause");
return 0;
}

```

*Алгоритм:*





Для побудови алгоритму було використано сайт: <https://www.yworks.com/yed-live>

*Результати роботи програми:*

```

D:\Projects\Sem3_Algorithms\ x + v
Введіть список чисел (для завершення введіть: "999" і натисніть Enter):
999

Список чисел:
999

=== Статистика ===
Загальна кількість елементів: 1
Додатних чисел: 1 (100.00%)
Від'ємних чисел: 0 (0.00%)
Нулів: 0 (0.00%)

Press any key to continue . . .
  
```

```
D:\Projects\Sem3_Algorithms/ x + v
Введіть список чисел (для завершення введіть: "999" і натисніть Enter):
-7 12 48 5 0 12 0 -100 1000 15 -5 -54 -7 12 34 65 999 2

Список чисел:
-7 12 48 5 0 12 0 -100 1000 15 -5 -54 -7 12 34 65 999

=== Статистика ===
Загальна кількість елементів: 17
Додатних чисел: 10 (58.82%)
Від'ємних чисел: 5 (29.41%)
Нулів: 2 (11.76%)

Press any key to continue . . . |
```

```
D:\Projects\Sem3_Algorithms/ x + v
Введіть список чисел (для завершення введіть: "999" і натисніть Enter):
17 -28 -4 76 -2 15 -50 0 -70 -35 12 0 999

Список чисел:
17 -28 -4 76 -2 15 -50 0 -70 -35 12 0 999

=== Статистика ===
Загальна кількість елементів: 13
Додатних чисел: 5 (38.46%)
Від'ємних чисел: 6 (46.15%)
Нулів: 2 (15.38%)

Press any key to continue . . . |
```

**Висновок.** У ході виконання роботи ми навчилися створювати лінійні списки, додавати елементи в кінець списку, реалізовувати підрахунок кількості додатних, від'ємних і нульових елементів та обчислювати їхній процентний вміст. Програма коректно працює для різних наборів вхідних даних, поставлену мету лабораторної роботи досягнуто.