

Програмування алгоритмічних структур

Лабораторна робота №5. Лінійні однозв'язні та двозв'язні списки.

Мета роботи:

Метою лабораторної роботи є ознайомлення з основами створення та особливостями роботи однозв'язних та двозв'язних списків, набуття навичок застосування структур даних даного виду для вирішення практичних задач.

Методичні вказівки:

Лабораторна робота спирається на знання і уміння, отримані при вивченні наступних питань лекції:

- Однозв'язний список є динамічною структурою даних, яка реалізує формування та обробку набору елементів.
- Елементи однозв'язного списку можуть бути відсортованими або не відсортованими.
- Однозв'язний список може зберігатися в оперативній пам'яті або у файлі.
- Двозв'язний лінійний список – це список, в якому попередній компонент посилається на наступний, а наступний на попередній.



ТЕОРЕТИЧНІ ВІДОМОСТІ [1/3]

Завдання лабораторної роботи:

Використовуючи на вибір студента однозв'язний або двозв'язний лінійний список розробити систему для виконання математичної операції над великими числами, визначеної індивідуальним варіантом.

Зверніть увагу:

Розглянуті приклади з розділу «Теоретичні відомості» можуть виконуватися по-різному, в залежності від компілятора або бітності вашої системи. Виконання відбувалося на x64-розрядній системі, на компіляторі MSVC.

Теоретичні відомості:

У мові програмування C++ визначено обмеження для значень цілочисельних констант. Діапазон значень для беззнакового типу даних **unsigned long long** встановлено від 0 до 18 446 744 073 709 551 615, включно:

```
unsigned long long a = 18446744073709551615; // допустимо
```

```
unsigned long long b = 18446744073709551616; // error, присвоїти значення неможливо
```

Якщо у математичній операції беруть участь числа, що є більшими за максимальне значення діапазону типу **unsigned long long** (**ULLONG_MAX**), то ця операція є недопустимою компілятором. Якщо таке число буде отримано у результаті виконання допустимих типом даних **unsigned long long** операндів, то результуюча змінна не зможе прийняти це число.

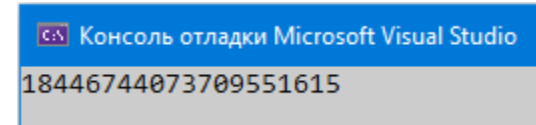


ТЕОРЕТИЧНІ ВІДОМОСТІ [2/3]

```
#include <iostream>
```

```
int main() {  
    unsigned long long a = 18446744073709551614;  
    unsigned long long b = 1;  
    unsigned long long c = a + b;  
  
    std::cout << c << std::endl;  
  
    return 0;  
}
```

run



Консоль отладки Microsoft Visual Studio
18446744073709551615

У наведеному прикладі, змінна “**c**” ініціалізується максимально допустимим значенням для типу **unsigned long long**. Обидва операнди “**a**” та “**b**” також є допустимими для присвоювання змінній такого типу.



ТЕОРЕТИЧНІ ВІДОМОСТІ [3/3]

```
#include <iostream>
```

```
int main() {  
    unsigned long long a = 18446744073709551615;  
    unsigned long long b = 5;  
    unsigned long long c = a + b;  
  
    std::cout << c << std::endl;  
  
    return 0;  
}
```

run



Консоль отладки Microsoft Visual Studio
4

Тепер змінна “**c**” ініціалізується недопустимим значенням для типу **unsigned long long** – воно перевищує максимально допустиме. У результаті отримуємо некоректне виконання математичної операції.

Змінна “**c**” ініціалізуватиметься значенням за наступною формулою:

*** [необхідний результат суми] - ULONG_MAX - 1**

де (для наведеного прикладу), **[необхідний результат суми]** дорівнює значенню **18446744073709551620**, а константа **ULONG_MAX** дорівнює значенню **18446744073709551615**.

* Варто зазначити, що формула є лише ілюстрацією для спрощення розуміння отриманого у результаті такої операції числа. Механізм визначення результуючого числа при виконанні програми, звісно, реалізовано інакше.



ЗАВДАННЯ РОБОТИ [1/3]

Отже, використовуючи лінійні *однозв'язні* або *двозв'язні* списки (на вибір студента), необхідно реалізувати систему для виконання визначеної індивідуальним варіантом однієї з трьох математичних операцій: додавання, віднімання, множення.

Одним з варіантів виконання таких математичних операцій над великими числами є застосування лінійних однозв'язних і двозв'язних списків.

Механіка виконання математичних операцій додавання, віднімання, множення, застосовуючи списки схожа на виконання аналогічних операцій додавання, віднімання, множення *в стовпчик*.

$$\begin{array}{r} \overset{1}{5} \overset{1}{3} \overset{1}{8} 4 6 \\ + 2 8 9 3 7 \\ \hline 8 2 7 8 3 \end{array}$$

Рис. 1 Додавання в стовпчик

$$\begin{array}{r} \overset{\bullet}{8} \overset{10}{7} \overset{\bullet}{0} \overset{10}{3} 2 \\ - 5 4 5 2 9 \\ \hline 5 0 3 \end{array}$$

Рис. 2 Віднімання в стовпчик

$$\begin{array}{r} \times 1 8 5 4 \\ 2 3 7 \\ \hline + 1 2 9 7 8 \\ + 5 5 6 2 \\ + 3 7 0 8 \\ \hline 4 3 9 3 9 8 \end{array}$$

Рис. 3 Множення в стовпчик

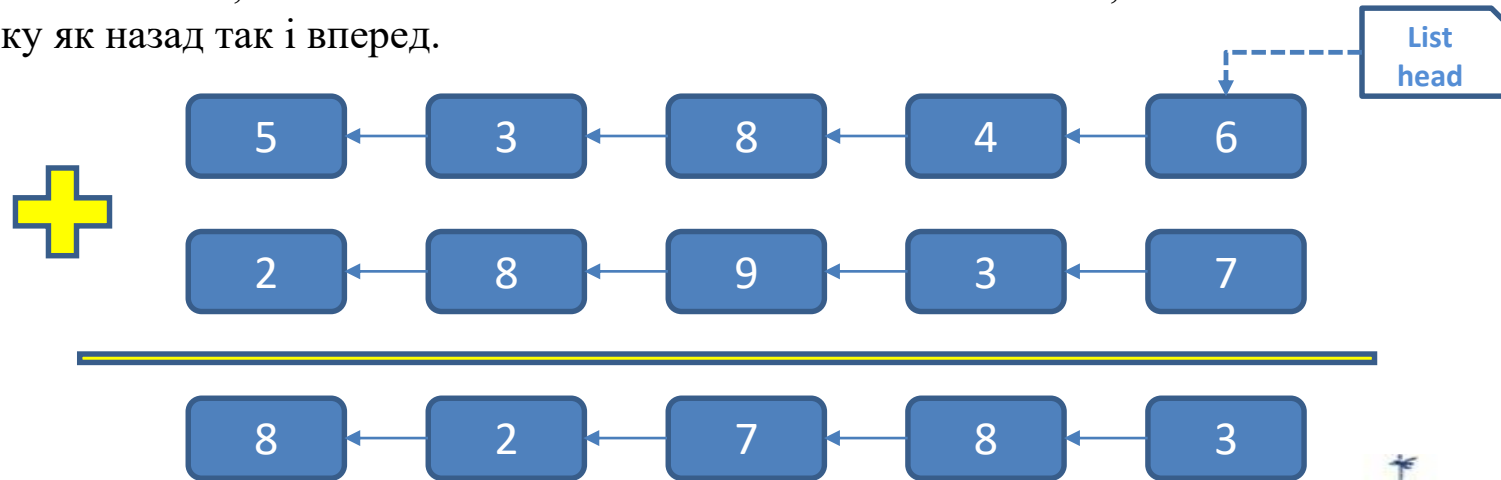


ЗАВДАННЯ РОБОТИ [2/3]

Для виконання поставленого завдання за допомогою списків необхідно кожен цифру вашого числа занести в окремий елемент списку, поєднуючи їх відповідними вказівниками (в залежності від типу списку або лише на наступний елемент, або і на наступний і на попередній).

- Для однозв'язного списку зручно заносити цифри числа у зворотному порядку, для спрощення процесу перегляду числа у списку, оскільки математичні операції “в стовпчик” виконуються починаючи з останньої цифри числа.
- Для двозв'язного списку такої проблеми не виникає, оскільки маючи вказівник на останній елемент, можна без перешкод рухатися за елементами списку як назад так і вперед.

Наприклад, реалізуючи операцію додавання з Рис. 1, для однозв'язного списку, заносимо усі цифри чисел у зворотному порядку в два списки. Під час процесу обчислення покроково створюється результуючий список.



ЗАВДАННЯ РОБОТИ [3/3]

Записувати великі числа у список можна використовуючи файлове введення (file input), строки (string), конвертуючи посимвольно елементи у числовий тип даних. Тобто, при обчисленнях не дозволяється зберігати операнди та результуючі числа у змінних. Числові дані використовуються лише для оперування **цифрами** ваших чисел

Приклад виконання операції додавання над великими числами:

Для прикладу було взято невиконувану математичну операцію зі Слайду 4: “**18446744073709551615 + 5**”
Зверніть увагу, елементи списку виведено у тому порядку, в якому вони заносилися, відповідно, щоб прочитати число, слід читати його з кінця списку.

```
Консоль отладки Microsoft Visual Studio

5->1->6->1->5->5->9->0->7->3->7->0->4->4->7->6->4->4->8->1->
+
5->
-----
0->2->6->1->5->5->9->0->7->3->7->0->4->4->7->6->4->4->8->1->

D:\КПИ\Algorithms\L5\Practice6\Debug\Practice6.exe (процесс 31600) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```



ІНДИВІДУАЛЬНІ ВАРІАНТИ. Група ТР-22

№	Математична операція	№	Математична операція	№	Математична операція
1.	Додавання	11.	Віднімання	21.	Додавання
2.	Множення	12.	Додавання	22.	Множення
3.	Віднімання	13.	Множення	23.	Додавання
4.	Множення	14.	Додавання	24.	Віднімання
5.	Додавання	15.	Додавання	25.	Множення
6.	Додавання	16.	Віднімання	26.	Додавання
7.	Множення	17.	Множення	27.	Множення
8.	Віднімання	18.	Додавання	28.	Віднімання
9.	Додавання	19.	Віднімання	29.	Додавання
10.	Множення	20.	Множення	30.	Віднімання

