

## Лабораторная работа № 1.8

### 1. Лінійний список (у вигляді масивів)

#### Задача:

Створюється структура даних лінійний однозв'язний список з необхідним набором операцій:

- створити порожній список;
- додати елемент до списку;
- додати елемент після заданого;
- видалити елемент зі списку;
- знайти елемент у списку;
- сортування елементів списку;
- додати елемент в відсортований список;
- перегляд всіх елементів списку;
- перегляд всіх елементів списку по циклу.

Під структуру даних виділяється неперервна ділянка пам'яті.

#### Вказівка:

При роботі зі створеною структурою необхідно аналізувати ситуацію, коли число елементів в списку перевищить допустиму кількість. У цій ситуації необхідно виділяти додатково необхідну кількість пам'яті, якщо це можливо. Детальний опис лінійного списку і роботи з ним можна знайти в [1,2].

### 2. Лінійний список (за допомогою вказівників/показників)

#### Задача:

Створюється структура даних лінійний однозв'язний список з необхідним набором операцій:

- створити порожній список;
- додати елемент до списку;
- додати елемент до списку після заданого елементу;
- видалити елемент зі списку;
- знайти елемент у списку;
- сортування елементів списку;
- додати елемент в відсортований список;
- перегляд всіх елементів списку.

Структура даних створюється з використанням вказівників/показників.

#### Вказівка:

При роботі зі створеною структурою необхідно в разі видалення елемента звільняти виділену під його зберігання пам'ять. Детальний опис лінійного списку і роботи з ним можна знайти в [1,2].

### **3. Структура даних нелінійний список**

#### Задача:

Створити мультисписок. Пов'язувати вузли можна за допомогою двох полів-показчиків в двох незалежних списках, по одному на кожне поле. Наприклад, одне поле пов'язує вузли в порядку їх створення. Друге поле, можливо, в відсортованому порядку. Це дозволяє вибрати порядок перегляду елементів.

#### Вказівка:

За допомогою генератора випадкових чисел створити мультисписок зі 100 елементів. Впорядкувати його за зростанням. Мати можливість переглядати елементи в порядку зростання і порядку генерації. Детальний опис нелінійного списку і роботи з ним можна знайти в [1,2].

### **СПИСОК ЛІТЕРАТУРИ**

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М: МЦНМО, 1999.
2. Седжвик Р. Фундаментальные алгоритмы на С++. Часть 5. – К: Издательство «ДиаСофт», 2002-4968с.