МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №10

з дисципліни
«Алгоритмізація та програмування»

Виконав:
студент групи КН-114
Добрій Назарій

Викладач:
Мочурад Л.І

Львів – 2019 р.

Тема: "Інформаційні динамічні структури"

Мета: Знайомство з динамічними інформаційними структурами на прикладі одно- і двонаправлених списків.

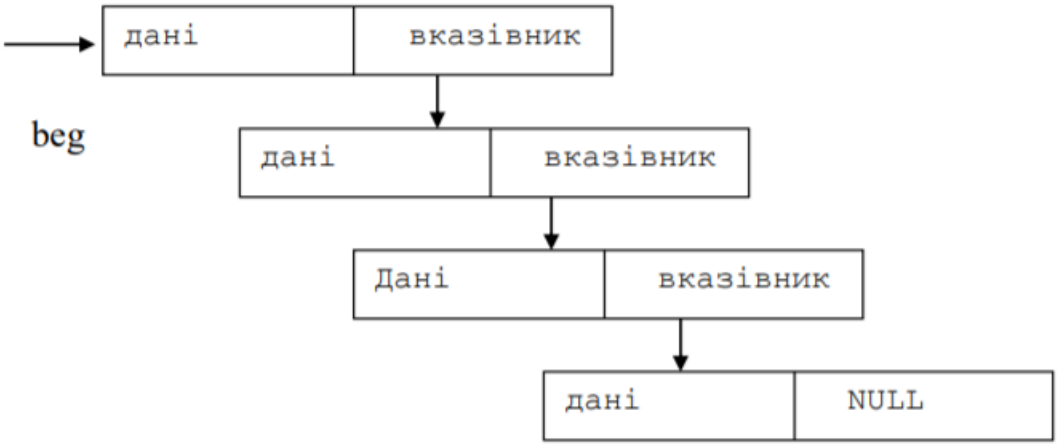
1. Короткі теоретичні відомості

У багатьох завданнях потрібно використовувати дані в яких конфігурація, розміри, склад можуть змінюватися в процесі виконання програми. Для їхнього представлення використовують динамічні інформаційні структури.

Найбільш проста інформаційна структура - це однозв'язний список, елементами якого є об'єкти структурного типу. Наприклад

```
struct ім'я_структурного_типу
{
елементи_структури;
struct ім'я_структурного_типу *вказівник;
}
```

У кожну структуру такого типу входить вказівник на об'єкт того ж типу, як і структура, що визначається.



2. Постановка завдання

Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом.

Для кожного варіанту розробити такі функції:

1. Створення списку.
2. Додавання елемента в список (у відповідності зі своїм варіантом).
3. Знищення елемента зі списку (у відповідності зі своїм варіантом).
4. Друк списку.
5. Запис списку у файл.
6. Знищення списку.
7. Відновлення списку з файлу.

Порядок виконання роботи

1. Написати функцію для створення списку. Функція може створювати порожній список, а потім додавати в нього елементи.
2. Написати функцію для друку списку. Функція повинна передбачати вивід повідомлення, якщо список порожній.
3. Написати функції для знищення й додавання елементів списку у відповідності зі своїм варіантом.

---

4. Виконати зміни в списку й друк списку після кожної зміни.
5. Написати функцію для запису списку у файл.
6. Написати функцію для знищення списку.
7. Записати список у файл, знищити його й виконати друк (при друці повинне бути видане повідомлення "Список порожній").
8. Написати функцію для відновлення списку з файлу.
9. Відновити список і роздрукувати його.
10. Знищити список.

Варіант 6

6. Записи в лінійному списку містять ключове поле типу int. Сформувати двонаправлений список. Знищити з нього елемент із заданим номером, додати елемент у початок списку.

Програмна реалізація:

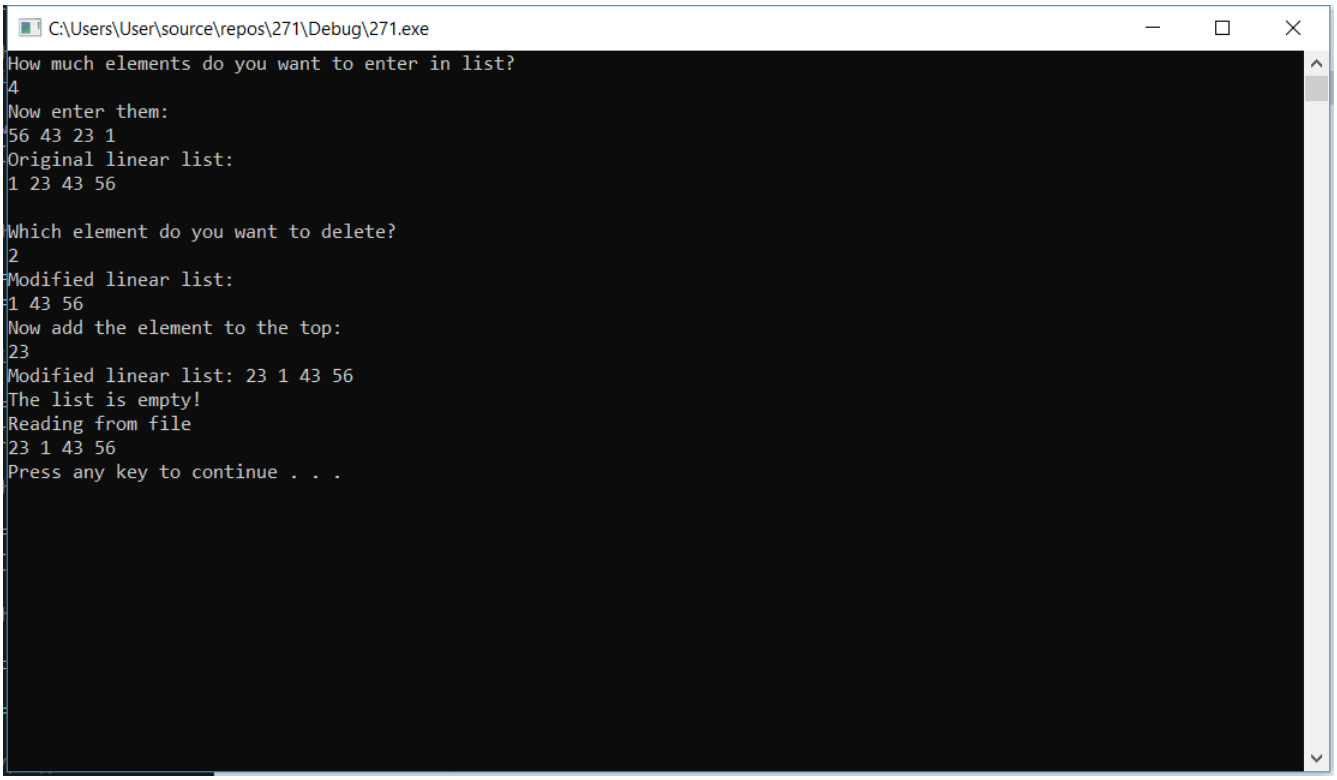
```cpp
#include <iostream>
#include <fstream>

using namespace std;

// Створюємо двусвязний список
struct Node {
    int data;
    Node* next;
    Node* prev;
};

//починаємо з порожнього списку
Node* head = NULL;

void deleteNode(Node** head, int position)
{
    // якщо список порожній
    if (*head == NULL)
    {
        return;
    }

    Node* temp = *head;

    if (position == 0)
    {
        *head = temp->next;
        free(temp);
        return;
    }

    for (int i = 0; temp != NULL && i < position - 1; i++)
    {
        temp = temp->next;
    }

    if (temp == NULL || temp->next == NULL)
    {
```

```cpp
    if (temp == NULL || temp->next == NULL)
    {
        return;
    }

    Node* next = temp->next->next;

    free(temp->next);  // звільняємо пам'ять

    temp->next = next;
}

void push_back(Node** head, int data)
{
    Node* new_node = new Node;
    new_node->data = data;
    new_node->prev = NULL;
    new_node->next = (*head);

    if ((*head) != NULL)
    {
        (*head)->prev = new_node;
    }

    (*head) = new_node;
}

void printList(Node* node)
{
    while (node != NULL)
    {
        cout << node->data << " ";
        node = node->next;
    }
    cout << endl;
}
void OutputToFile(Node** head)
{
```

```
}
void OutputToFile(Node** head)
{
    ofstream fout("file.txt");
    if (fout)
    {
        Node* temp = *head; //оголошуємо вказівник який вказує на початок

        while (temp != NULL) //поки по адресу щось є
        {
            //вводимо структуру в файл
            fout << temp->data << "  ";

            temp = temp->next; //вказуємо на наступний елемент списку
        }
        if (temp == NULL)
        {
            cout << "The list is empty!" << endl;
        }

        fout.close();
    }
    else
    {
        cout << "Error creating file!" << endl;
    }
}


int main()
{
    cout << "How much elements do you want to enter in list?" << endl;

    int numberNode, element, countDeleteElement, add;
    cin >> numberNode;

    cout << "Now enter them:" << endl;
    //вводимо елементи
    for (int i = 0; i < numberNode; i++) {
    cout << "Now enter them:" << endl;
    //вводимо елементи
    for (int i = 0; i < numberNode; i++) {
        cin >> element;
        push_back(&head, element);
    }

    cout << "Original linear list: " << endl;
    printList(head);

    cout << "\nWhich element do you want to delete?" << endl;
    cin >> countDeleteElement;
    deleteNode(&head, countDeleteElement - 1);

    cout << "Modified linear list: " << endl;
    printList(head);

    cout << "Now add the element to the top: " << endl;
    cin >> add;
    push_back(&head, add);

    cout << "Modified linear list: ";

    printList(head);
    OutputToFile(&head);

    //видаляємо список
    for (int i = 0; i < numberNode; i++) {
        deleteNode(&head, 0);
    }
    //зчитуємо зі списку в файл(відновлюємо список)
    cout << "Reading from file" << endl;

    fstream fs("file.txt");

    if (!fs)
    {
        cout << "File not found\n";
    }
```

```
148     {
149         cout << "File not found\n";
150     }
151     int count = 0;
152
153     while (!fs.eof())
154     {
155         fs >> element;
156         count++;
157     }
158
159     fs.clear();
160     fs.seekg(0);
161
162     int* mas = new int[count];
163
164     for (int i = 0;i < count;i++) // записуємо елементи файлу в масив
165     {
166         fs >> mas[i];
167     }
168
169     for (int i = 0; i < count - 1; i++)
170     {
171         cout << mas[i] << ' ';
172     }
173
174     cout << "\n";
175
176     fs.close();
177
178     //видаляємо список знову
179     for (int i = 0; i < numberNode; i++) {
180         deleteNode(&head, 0);
181     }
182
183     system("pause");
184     return 0;
185 }
```

Вивід програми:



Висновок: я познайомився з динамічними інформаційними структурами на прикладі одно- і двонаправлених списків.