

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота № 2.3
з дисципліни
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-22
Коваленко Михайло Володимирович
номер у списку групи: 11

Перевірила:

Молчанова А. А.

Київ 2023

ЛАБОРАТОРНА РОБОТА №2.2.
ЗВ'ЯЗАНІ ДИНАМІЧНІ СТРУКТУРИ ДАНИХ.
СПИСКИ

Мета лабораторної роботи

Метою лабораторної роботи №2.2 є засвоєння теоретичного матеріалу та набуття практичного досвіду використання зв'язаних динамічних структур даних у вигляді одно- та двозв'язних списків при складанні різних алгоритмів.

Постановка задачі

1. Створити список з n ($n > 0$) елементів (n вводиться з клавіатури), якщо інша кількість елементів не вказана у конкретному завданні.
2. Тип ключів (інформаційних полів) задано за варіантом.
3. Значення елементів списку взяти самостійно такими, щоб можна було продемонструвати коректність роботи алгоритму програми. Введення значень елементів списку можна виконати довільним способом (випадкові числа, формування значень за формулою, введення з файлу чи з клавіатури).
4. Вид списку (черга, стек, дек, прямий однозв'язний лінійний список, обернений однозв'язний лінійний список, двозв'язний лінійний список, однозв'язний кільцевий список, двозв'язний кільцевий список) вибрати самостійно з метою найбільш доцільного рішення поставленої за варіантом задачі.
5. Виконати над створеним списком дії, вказані за варіантом, та коректне звільнення пам'яті списку.
6. При виконанні заданих дій, виводі значень елементів та звільненні пам'яті списку вважати, що довжина списку (кількість елементів n чи $2n$) невідома на момент виконання цих дій.
7. Повторювані частини алгоритму необхідно оформити у вигляді процедур або функцій (для створення, обробки, виведення та звільнення пам'яті списків) з передачею списку за допомогою параметра(ів).

Завдання варіанту

Варіант 11

Ключами елементів списку є латинські літери. Відсортувати елементи списку у лексикографічному порядку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»), методом вставки.

Текст програми:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
    char key;
    struct Node* next;
} Node;

void insertElement(Node** head, char key) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->key = key;
    newNode->next = NULL;
    if (*head == NULL) {
        *head = newNode;
        return;
    }
    Node* curr = *head;
    while (curr->next != NULL) {
        curr = curr->next;
    }
    curr->next = newNode;
}

void sortedInsert(Node** head, Node* newNode) {
    if (*head == NULL || newNode->key <= (*head)->key) {
        newNode->next = *head;
        *head = newNode;
        return;
    }

    Node* curr = *head;
    while (curr->next != NULL && newNode->key > curr->next->key) {
        curr = curr->next;
    }
    newNode->next = curr->next;
    curr->next = newNode;
}

void sortList(Node** head) {
    Node* sortedList = NULL;
    Node* curr = *head;
    while (curr != NULL) {
        Node* nextNode = curr->next;
        sortedInsert(&sortedList, curr);
        curr = nextNode;
    }
}
```

```

    *head = sortedList;
}

void printList(Node* head) {
    Node* curr = head;
    while (curr != NULL) {
        printf("%c ", curr->key);
        curr = curr->next;
    }
    printf("\n");
}

void freeList(Node* head) {
    Node* curr = head;
    while (curr != NULL) {
        Node* nextNode = curr->next;
        free(curr);
        curr = nextNode;
    }
}

int main() {
    Node* head = NULL;
    int n;
    char key;
    printf("Input number ov elements: ");
    scanf("%d", &n);

    printf("Input elements:\n");
    for (int i = 0; i < n; i++) {
        printf("Element %d: ", i + 1);
        scanf(" %c", &key);
        insertElement(&head, key);
    }

    printf("Default list: ");
    printList(head);
    sortList(&head);
    printf("Sorted list: ");
    printList(head);
    freeList(head);
    return 0;
}

```

Тестування програми:

```

lab2semester2 x
D:\Coding\CLionProjects\lab2semester2\cmake-build-debug\lab2semester2.exe
Input number ov elements: 4
Input elements:
Element 1: a
Element 2: f
Element 3: s
Element 4: r
Default list: a f s r
Sorted list: a f r s

Process finished with exit code 0

```

```
lab2semester2 x
D:\Coding\CLionProjects\lab2semester2\cmake-build-debug\lab2semester2.exe
Input number ov elements:4
Input elements:
Element 1:d
Element 2:a
Element 3:c
Element 4:b
Default list: d b c a
Sorted list: a b c d

Process finished with exit code 0
```

```
lab2semester2 x
D:\Coding\CLionProjects\lab2semester2\cmake-build-debug\lab2semester2.exe
Input number ov elements:5
Input elements:
Element 1:t
Element 2:z
Element 3:f
Element 4:l
Element 5:a
Default list: t z f l a
Sorted list: a f l t z

Process finished with exit code 0
```