

**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**  
з дисципліни  
«Алгоритми і структури даних»

Виконав:

Студент групи ІМ-41

Номер у списку групи: 7

Перевірив:

Сергієнко А. М.

Київ 2025

## Завдання

1. Створити список з  $n$  ( $n > 0$ ) елементів ( $n$  вводиться з клавіатури), якщо інша кількість елементів не вказана у конкретному завданні за варіантом.
2. Тип ключів (інформаційних полів) задано за варіантом.
3. 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 розв'язку поставленої за варіантом задачі.
4. Створити функції (або процедури) для роботи зі списком (для створення, обробки, додавання чи видалення елементів, виводу даних зі списку в консоль, звільнення пам'яті тощо).
5. 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).
6. Виконати над створеним списком дії, вказані за варіантом, та коректне звільнення пам'яті списку.
7. При виконанні заданих дій, виводі значень елементів та звільненні пам'яті списку вважати, що довжина списку (кількість елементів) невідома на момент виконання цих дій. Тобто, не дозволяється зберігати довжину списку як константу, змінну чи додаткове поле.

## Варіант 7:

Варіант № 7

Ключами елементів списку є цілі числа. Кількість елементів списку повинна дорівнювати  $2n$ . Обчислити значення виразу:  $a_1 \cdot a_{2n} + a_2 \cdot a_{2n-1} + \dots + a_n \cdot a_{n+1}$ , де  $a_i$  —  $i$ -й елемент списку.

## Текст програми

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

typedef struct ListNode {
    int data;
```

```

    struct ListNode* next;

    struct ListNode* prev;
} ListNode;

ListNode* createNode(const int data)
{
    ListNode* newNode = (ListNode*)malloc(sizeof(ListNode));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

void pushBack(ListNode** tail, const int data)
{
    ListNode* newNode = createNode(data);

    if (*tail == NULL) {
        *tail = newNode;
        return;
    }

    ListNode* temp = *tail;
    temp->next = newNode;
    newNode->prev = temp;

    *tail = newNode;
}

```

```
void printListForward(ListNode* head)
{
    ListNode* temp = head;
    printf("List: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

void clearListFromHead(ListNode* head) {
    ListNode* temp = head;
    while (temp != NULL) {
        ListNode* next = temp->next;
        free(temp);
        temp = next;
    }
}

int main(void) {
    int n;
    printf("Enter n: ");
    scanf("%d", &n);

    ListNode* head = NULL;
    pushBack(&head, 1);

    ListNode* tail = head;
```

```
for (int i = 2; i ≤ 2 * n; i++) {
    pushBack(&tail, i);
}

printListForward(head);

ListNode* start = head;

int result = 0;

for (int i = 0; i < n; i++) {
    result += start→data * tail→data;

    start = start→next;
    tail = tail→prev;
}

printf("Result: %d\n", result);

clearListFromHead(head);

return 0;
}
```

**Тести програми:**

```
/Users/illia/CLionProjects/asd/lab-new-2/buildDir/lab_new_2
Enter n: 4
List: 1 2 3 4 5 6 7 8
Result: 60
```

```
/Users/illia/CLionProjects/asd/lab-new-2/buildDir/lab_new_2
Enter n: 5
List: 1 2 3 4 5 6 7 8 9 10
Result: 110
```

Process finished with exit code 0

```
/Users/illia/CLionProjects/asd/lab-new-2/buildDir/lab_new_2
Enter n: 6
List: 1 2 3 4 5 6 7 8 9 10 11 12
Result: 182
```

Process finished with exit code 0

```
/Users/illia/CLionProjects/asd/lab-new-2/buildDir/lab_new_2
Enter n: 7
List: 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Result: 280
```

Process finished with exit code 0

```
/Users/illia/CLionProjects/asd/lab-new-2/buildDir/lab_new_2
Enter n: 8
List: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Result: 408

Process finished with exit code 0
```

## Висновок:

Під час виконання роботи навчився створювати і використовувати двозв'язний лінійний список, динамічно алокуючи пам'ять (не забуваючи про її звільнення в кінці для запобігання memory leak). Він є оптимальною структурою даних для цього завдання, бо окрім простого і швидкого (складність  $O(1)$ ) додавання нових елементів можна легко пересуватись одночасно вперед і назад по списку.