

**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 № 2.2**

з дисципліни  
«Алгоритми і структури даних»

Виконав

Студент групи ІП-03  
Пашковський Євгеній Сергійович  
номер у списку групи: 17

Перевірила:

Сергієнко А. А.

## Завдання

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  $2n$ ) **невідомо** 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).

### Варіант 17:

Ключами елементів списку є дійсні числа. Обчислити значення виразу:  
 $a_1 a_n + a_2 a_{n-1} + \dots + a_n a_1$ , де  $a_i$  –  $i$ -й елемент списку.

## Текст програми

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

typedef struct linked_list
{
    double data;
    struct linked_list * next_p;
    struct linked_list * prev_p;
} l_list;

l_list * l_list_init(double item)
{
    l_list *l_p;
    l_p = malloc(sizeof(struct linked_list));

    l_p->data = item;
    l_p->next_p = l_p;
    l_p->prev_p = NULL;
    return l_p;
}

l_list * get_last_node(l_list *l_p)
{
    l_list *this_p = l_p;
    while(this_p->prev_p)
    {
        this_p = this_p->prev_p;
    }
    return this_p;
}

l_list * add_node(l_list *l_p, double item)
{
    l_list *node_p;
    node_p = malloc(sizeof(struct linked_list));
    l_p->next_p = node_p;

    node_p->data = item;
    node_p->next_p = get_last_node(l_p);
    node_p->prev_p = l_p;
    return node_p;
}

//remove node from begining
l_list * unshift_node(l_list *l_p)
{
    l_list *new_l_p = l_p->prev_p;
    new_l_p->next_p = l_p->next_p;
    free(l_p);
    return new_l_p;
}
```

```

//remove node from end
l_list * pop_node(l_list *l_p)
{
    if (l_p->next_p != l_p)
    {
        l_list *last_node = l_p->next_p;

        last_node->next_p->prev_p = NULL;

        l_p->next_p = last_node->next_p;

        free(last_node);
        return l_p;
    } else {
        free(l_p);
        return NULL;
    }
}

int main()
{
    system("cls");
    unsigned int n = 0;

    while(n <= 0 || n % 2 != 0)
    {
        printf("n (n %% 2 = 0; n > 0): ");
        scanf("%u", &n);
    }

    l_list *list_p;
    for (unsigned int i = n; i >= 1; i--)
    {
        double a;
        printf("a%u: ", i);
        scanf("%lf", &a);

        if (i == n)
        {
            //initialization
            list_p = l_list_init(a);
        } else {
            //fulfilling
            list_p = add_node(list_p, a);
        }
    }
    //operations
    double result = 0;
    while (list_p) {
        result += 2 * (list_p->data) * (list_p->next_p->data);
        list_p = unshift_node(list_p);
        list_p = pop_node(list_p);
    }
    printf("Result: %lf\n", result);
    system("pause");
    return 0;
}

```

# Тестування програми

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 4
a4: 4
a3: 3
a2: 2
a1: 1
Result: 20.000000
Для продолжения нажмите любую клавишу . . . |
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 10
a10: 10
a9: 9
a8: 8
a7: 7
a6: 6
a5: 5
a4: 4
a3: 3
a2: 2
a1: 1
Result: 220.000000
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 4
a4: 3
a3: 2
a2: 1
a1: 1
Result: 10.000000
Для продолжения нажмите любую клавишу . . . |
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 2
a2: 3
a1: 1
Result: 6.000000
Для продолжения нажмите любую клавишу . . . |
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 1
n (n % 2 = 0; n > 0): 2
a2: 11
a1: 1
Result: 22.000000
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 4
a4: 2.3
a3: 1
a2: 4.2
a1: 1
Result: 13.000000
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 6
a6: 2.1
a5: 2.6
a4: 1.2
a3: 1
a2: 1
a1: 2
Result: 16.000000
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\eeegen\OneDrive\Рабочий стол\ДЗ\Лабы\АСД\2 семестр\2.2\Programs\Lab2.2-2.exe
n (n % 2 = 0; n > 0): 2
a2: 1.5215
a1: 0
Result: 0.000000
Для продолжения нажмите любую клавишу . . . |
```