

**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 ІМ-34  
Никифоров Артем Михайлович  
Номер у списку групи: 16

Перевірила:  
Молчанова А. А.

### **Постановка задачі:**

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). 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.

### **Завдання для конкретного варіанту:**

#### **Варіант № 16**

Ключами елементів списку є дійсні числа. Розширити список, дописавши в його кінець свої ж елементи, але у оберненому порядку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

**Текст програми:**

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

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

```
typedef struct linked_list {
```

```
    int info;
```

```
    struct linked_list* next_p;
```

```
} l_list;
```

```
l_list* l_list_init(int item) {
```

```
    l_list *l_p;
```

```
    l_p = malloc(sizeof(struct linked_list));
```

```
    l_p->info = item;
```

```
    l_p->next_p = NULL;
```

```
    return l_p;
```

```
}
```

```
l_list* add_list(l_list* head, int item) {
```

```
    if (head == NULL) {
```

```
        return l_list_init(item);
```

```
    }
```

```
    l_list* current = head;
```

```
    while (current->next_p != NULL) {
```

```
        current = current->next_p;
```

```
    }
```

```
    current->next_p = l_list_init(item);
```

```
    return head;
```

```
}
```

```

void free_list(l_list* head) {
    l_list* temp;
    while (head != NULL) {
        temp = head;
        head = head->next_p;
        free(temp);
    }
}

void print_list(l_list* head) {
    l_list* reversed_head = NULL;
    printf("Linked List: ");
    while (head != NULL) {
        printf("%d ", head->info);
        l_list* temp = l_list_init(head->info);
        temp->next_p = reversed_head;
        reversed_head = temp;
        head = head->next_p;
    }
    while (reversed_head != NULL) {
        printf("%d ", reversed_head->info);
        reversed_head = reversed_head->next_p;
    }
}

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

```

```
l_list* head = NULL;
printf("Enter %d elements:\n", n);
for (int i = 0; i < n; i++) {
    int item;
    scanf("%d", &item);
    head = add_list(head, item);
}
print_list(head);
free_list(head);
return 0;
}
```

## Скріншоти тестування програми:

```
Enter the number of elements:5
Enter 5 elements:
1
2
3
4
5
Linked List: 1 2 3 4 5 5 4 3 2 1

Process finished with exit code 0
```

```
Enter the number of elements:10
Enter 10 elements:
-3
-2
-1
0
1
2
3
4
5
999
Linked List: -3 -2 -1 0 1 2 3 4 5 999 999 5 4 3 2 1 0 -1 -2 -3

Process finished with exit code 0
|
```

```
Enter the number of elements:10
Enter 10 elements:
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
Linked List: -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1
Process finished with exit code 0
```

**Висновок:**

Виконуючи цю лабораторну роботу, я засвоїв теоретичний матеріал та набув практичного досвіду використання динамічних структур даних у вигляді одно- та двов'язних списків при складанні різних алгоритмів.