

**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-23  
Юхименко Станіслав Романович  
номер у списку групи: 28

Перевірила:

Молчанова А. А.

Київ 2023

## 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).

### Варіант 28

Ключами елементів списку є цілі ненульові числа. Кількість елементів списку  $n$  повинна бути кратною 10-ти, а елементи у початковому списку розташовуватись із чергуванням знаків. Перекомпонувати список, змінюючи порядок чисел всередині кожного десятка елементів так, щоб спочатку йшли від'ємні числа цього десятка елементів, а за ними – додатні, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

## Текст програми main.c

```
#define _CRTDBG_MAP_ALLOC
#include <crtdbg.h>
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
    int list;
    struct Node* next;
} Node;

void insertNode(Node** root, int list) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->list = list;
    newNode->next = NULL;

    if (*root == NULL) {
        *root = newNode;
    }
    else {
        Node* current = *root;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}

void printList(Node* root) {
    Node* current = root;
    while (current != NULL) {
        printf("%d ", current->list);
        current = current->next;
    }
    printf("\n");
}

void reorderList(Node** root) {
    Node* current = *root;
    Node* prev = NULL;

    while (current != NULL) {
        int i = 0;
        Node* negativeFirst = NULL;
        Node* negativeLast = NULL;
        Node* positiveFirst = NULL;
        Node* positiveLast = NULL;

        while (current != NULL && i < 10) {
            Node* nextNode = current->next;

            if (current->list < 0) {
                if (negativeFirst == NULL) {
                    negativeFirst = current;
                    negativeLast = current;
                }
                else {
                    negativeLast->next = current;
                    negativeLast = current;
                }
            }
            else {
                if (positiveFirst == NULL) {
                    positiveFirst = current;
                    positiveLast = current;
                }
            }
            current = nextNode;
            i++;
        }
    }
}
```

```

        else {
            positiveLast->next = current;
            positiveLast = current;
        }
    }

    current->next = NULL;
    current = nextNode;
    i++;
}

if (negativeFirst != NULL) {
    if (prev != NULL) {
        prev->next = negativeFirst;
    }
    else {
        *root = negativeFirst;
    }
    if (positiveFirst != NULL) {
        negativeLast->next = positiveFirst;
        prev = positiveLast;
    }
    else {
        prev = negativeLast;
    }
}
else {
    if (prev != NULL) {
        prev->next = positiveFirst;
    }
    else {
        *root = positiveFirst;
    }
    prev = positiveLast;
}
}

}

void deallocateList(Node** root) {
    Node* current = *root;
    while (current != NULL) {
        Node* nextNode = current->next;
        free(current);
        current = nextNode;
    }
    *root = NULL;
}

int main() {
    srand(time(NULL));

    int n;
    printf("Enter n (n must be a multiple of 10): ");
    scanf_s("%d", &n);

    if (n % 10 != 0) {
        printf("Error: n must be a multiple of 10.\n");
        return 1;
    }

    Node* root = NULL;

    printf("Random List: \t");
    for (int i = 0; i < n; i++) {
        int sign = (i % 2 == 0) ? 1 : -1;
        int list = ((rand() % 100) + 1) * sign;
        insertNode(&root, list);
    }
}

```

```

}

printList(root);

reorderList(&root);

printf("Reordered list: ");
printList(root);

deallocateList(&root);

_CrtDumpMemoryLeaks(); // for debugging memory leak (from #define
_CRTDBG_MAP_ALLOC and crtDBG.h)

return 0;
}

```

## Скріншоти тестування програми

```

Microsoft Visual Studio Debug Console
Enter n (n must be a multiple of 10): 20
Random list: 42 -94 59 -73 97 -12 15 -73 96 -36 40 -90 36 -64 63 -4 65 -57 47 -7
Reordered list: -94 -73 -12 -73 -36 42 59 97 15 96 -90 -64 -4 -57 -7 40 36 63 65 47

C:\Users\ADMIN\source\repos\Lab_2_2_2\x64\Debug\Lab_2_2_2.exe (process 43928) exited with code 0.
Press any key to close this window . . .

```

```

Microsoft Visual Studio Debug Console
Enter n (n must be a multiple of 10): 10
Random list: 57 -73 13 -21 59 -76 46 -22 78 -29
Reordered list: -73 -21 -76 -22 -29 57 13 59 46 78

C:\Users\ADMIN\source\repos\Lab_2_2_2\x64\Debug\Lab_2_2_2.exe (process 18556) exited with code 0.
Press any key to close this window . . .

```

```

Microsoft Visual Studio Debug Console
Enter n (n must be a multiple of 10): 30
Random list: 91 -73 48 -77 72 -32 85 -76 93 -34 67 -37 22 -100 100 -11 48 -50 10 -10 12 -45 20 -53 67 -22 81 -56 87 -79
Reordered list: -73 -77 -32 -76 -34 91 48 72 85 93 -37 -100 -11 -50 -10 67 22 100 48 10 -45 -53 -22 -56 -79 12 20 67 81 87

C:\Users\ADMIN\source\repos\Lab_2_2_2\x64\Debug\Lab_2_2_2.exe (process 39904) exited with code 0.
Press any key to close this window . . .

```

```

Microsoft Visual Studio Debug Console
Enter n (n must be a multiple of 10): 5
Error: n must be a multiple of 10.

C:\Users\ADMIN\source\repos\Lab_2_2_2\x64\Debug\Lab_2_2_2.exe (process 12016) exited with code 1.
Press any key to close this window . . .

```

## Посилання на гітхаб репозиторій:

<https://github.com/stasyukhymenko/ASD-Lab-2-2.2.git>