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-04
Пащенко Дмитро Олексійович
номер у списку групи: 19

Перевірила:

Сергієнко А. А.

Київ 2021

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 19

Задано два списки, список S1 довжиною 2n елементів і список S2 довжиною n елементів. Ключами елементів обох списків є натуральні числа. Вставити список S2 у середину списку S1, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

Текст програми

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

typedef struct s_list {
    int num;

    struct s_list *next;
} t_list;

t_list *create_node(int set_num) {
    t_list *node = (t_list *)malloc(sizeof(t_list));

    node -> num = set_num;

    node -> next = NULL;
    return node;
}

void push_front(t_list **list, int set_num) {
    t_list *new_element = create_node(set_num);

    new_element -> next = *list;
    *list = new_element;
}

void insert(t_list *first, t_list *second) {
    t_list *curr = first;
    for (int i = 1; i < 2; i++) {
        curr = curr -> next;
    }

    t_list *temp = curr -> next;

    curr -> next = second;

    while (second -> next != NULL) {
        second = second->next;
    }

    second -> next = temp;
}

int main() {
    int n;
    printf("Enter n:");
    scanf("%d", &n);
    if (n > 0) {
        printf("n is %i\n", n);
```

```
        t_list *S1 = create_node(1);
        for (int i = 2; i < (2 * n) + 1; i++) {
            push_front(&S1, i);
        }

        t_list *S2 = create_node(100);
        for (int i = 101; i < n + 100; i++) {
            push_front(&S2, i);
        }

        insert(S1, S2);
        printf("\nS1 is now:\n");
        while (S1 != NULL) {
            printf("num is %d\n", S1 -> num);
            S1 = S1 -> next;
        }

    } else printf("n must be greater than 0");
    return 0;
}
```

Тестування

```
Enter n:1
n is 1

S1 is now:
num is 2
num is 1
num is 100
```

```
Enter n:2
n is 2

S1 is now:
num is 4
num is 3
num is 101
num is 100
num is 2
num is 1
```

```
Enter n:3
n is 3

S1 is now:
num is 6
num is 5
num is 102
num is 101
num is 100
num is 4
num is 3
num is 2
num is 1
```

```
Enter n:4
n is 4

S1 is now:
num is 8
num is 7
num is 103
num is 102
num is 101
num is 100
num is 6
num is 5
num is 4
num is 3
num is 2
num is 1
```

```
Enter n:5
n is 5

S1 is now:
num is 10
num is 9
num is 104
num is 103
num is 102
num is 101
num is 100
num is 8
num is 7
num is 6
num is 5
num is 4
num is 3
num is 2
num is 1
```