

**Звіт з лабораторної роботи  
за дисципліною "алгоритми і структури даних"  
студента групи ПА-17-1  
Панасенка Єгора Сергійовича  
Кафедра комп'ютерних технологій, фпм, дну  
2017-2018 навч.р.**

1. Постановка задачі:

а) Розробити абстрактний тип даних (АТД) «Лінійний одно зв'язаний список». На початку роботи програми список не містить елементів. Інтерфейс має включати такі операції:

1. Додавання  $n$  елементів в кінець списку. Передбачити 2 варіанти вводу елементів: 1) з клавіатури; 2) з файлу data.txt. Порядок формування файлу data.txt див. нижче.

2. Додавання одного елементу до списку.

1. для парних варіантів – на початок списку,

2. для непарних варіантів - в кінець списку.

Значення елементу для додавання вводити з клавіатури.

3. Видалення елементу із списку (парний)

1. для парних варіантів – за номером. Номер вводити з клавіатури. Використати операцію з п.4.

2. для непарних варіантів - за значенням. Значення елементу вводити з клавіатури. Видалити перший елемент з заданим значенням, який зустрівся. Використати операцію з п.5.

Результат вивести на екран.

4. Пошук елементу в списку за номером.

Номер вводити з клавіатури. Результат – значення елементу – вивести на екран.

5. Пошук елементу в списку за значенням.

Реалізувати алгоритм лінійного пошуку заданого елементу в списку. Відшукати перший елемент з заданим значенням, який зустрівся. Значення елементу для пошуку вводити з клавіатури. Результат – номер елементу в списку або повідомлення про відсутність елементу з таким значенням – вивести на екран.

6. Виведення списку на екран.

7. Виведення на екран поточної кількості елементів в списку.
8. Очищення списку.

Порядок формування файлу data.txt: Для формування файлу data.txt розробити окрему програму, яка за допомогою датчика випадкових чисел генерує необхідну кількість даних в певному діапазоні та записує їх у файл. Кількість та діапазон чисел вводити з клавіатури.

- b) Реалізувати дії згідно з варіантом, використовуючи розроблені операції зі списком.

1. Реалізувати алгоритм, який видаляє кожний другий елемент списку.

## 2. Опис структури програми та реалізованих функцій:

- a) Програма задає тип `element` з `struct element`, а також задає `struct element` з такими полями:

1. `value` – де зберігається значення одного елемента
2. `next` – де зберігається посилання на наступний елемент

- b) Програма має такі функції:

1. `void ign_other(FILE * input)`

1. Ігнорує непотрібні данні до кінця рядка. Наприклад якщо ми запросили одне число, а ввели число і якийсь текст то програма забере тільки число.

2. Аргументи:

1. `input` - вхідний потік.

3. Функція нічого не виводить.

2. `int get_number (FILE * input, FILE * output, char message[], int min, int max)`

1. Циклічно забирає число у проміжку з `min` до `max` з вхідного потоку доки не отримає потрібне число

2. Аргументи

1. `input` – вхідний потік даних

2. `output` – вихідний потік даних

3. `message` – повідомлення у якому повинен буди запит на число

4. `min` – мінімальне доступне значення

5. `max` – максимально доступне значення

3. Функція виводить число від `min` до `max`

3. `element * add_to_end(element * head, int value)`
  1. Додає один елемент у кінець списку
  2. Аргументи
    1. `head` – посилання на перший елемент
    2. `value` – значення яке буде у нового елемента
  3. Функція виводить посилання на перший елемент
4. `element * fill_list(FILE * input, element * head, int count)`
  1. Додає `count` елементів з потоку у кінець списку
  2. Аргументи
    1. `input` – вхідний потік даних
    2. `head` – посилання на перший елемент
    3. `count` – кількість елементів які потрібно забрати
  3. Функція виводить посилання на перший елемент
5. `char show_list (FILE * output, element * head)`
  1. Показує усі елементи списку в `output`
  2. Аргументи
    1. `output` – вихідний потік даних
    2. `head` – посилання на перший елемент
  3. Функція виводить 0 після успішного виконання, або 1 якщо список був пустий
6. `size_t count_list (element * head)`
  1. Рахує кількість елементів у списку
  2. Аргументи
    1. `head` – посилання на перший елемент
  3. Виводить кількість елементів
7. `element * find_by_key (element * head, int key)`
  1. Шукає елемент по номеру `key`
  2. Аргументи
    1. `head` – посилання на перший елемент
    2. `key` – номер потрібного елемента
  3. Посилання на елемент, або `NULL`, якщо елемент не знайдений

8. `int find_by_value (element * head, int value)`
  1. Шукає елемент за значенням `value`
  2. Аргументи
    1. `head` – посилання на перший елемент
    2. `value` – значення потрібного елемента
  3. Функція виводить номер елемента, або -1 якщо елемент не знайдений
9. `element * delete_node_by_value (element * head, int value)`
  1. Видаляє один елемент за значенням
  2. Аргументи
    1. `head` – посилання на перший елемент
    2. `value` – значення потрібного елемента
  3. Функція виводить посилання на перший елемент
10. `element * delete_node_by_key (element * head, int key)`
  1. Видаляє один елемент за номером
  2. Аргументи
    1. `head` – посилання на перший елемент
    2. `key` – номер потрібного елемента
  3. Функція виводить посилання на перший елемент
11. `void delete_even (element * head)`
  1. Видаляє елементи з парними номерами з квадратичною швидкістю
  2. Аргументи
    1. `head` – посилання на перший елемент
  3. Функція нічого не виводить
12. `void delete_even_fast (element * head)`
  1. Видаляє елементи з парними номерами з лінійною швидкістю
  2. Аргументи
    1. `head` – посилання на перший елемент
  3. Функція нічого не виводить
13. `element * delete_list (element * head)`
  1. Видаляє усі елементи
  2. Аргументи

1. head – посилання на перший елемент

3. Функція виводить посилання на перший елемент, а так як список пустий, то завжди виводить NULL

с) У головній функції виконуються такі дії:

1. Цикл доки не отримаємо запит на вихід програми

1. Вихід меню

2. Запит числа

3. За пунктом меню виконує потрібну функцію

4. Якщо не потрібно виходити з програми

1. Показує список

2. Запитує ENTER для продовження

3. Запитує данні доки не отримаємо ENTER

2. Видаляємо список

3. Виходимо з програми

d) У програмі datagen, яка генерує випадкові числа, виконуються такі дії:

1. Запит кількості елементів для генерування

2. Запит інтервалу допустимих значень

3. Заповнення data.txt випадковими числами

3. Код:

main.c

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

```
void ign_other(FILE * input) {
    char c = 0;
    while (c!='\n')
        c=fgetc(input);
}
```

```
int get_number (FILE * input, FILE * output, char message[], int min, int max) {
    int number = min-1;
    while (number < min || number > max) {
        fprintf(output,"%s:\n",message);
        fscanf(input,"%i",&number);
        ign_other(stdin);
        if (number < min || number > max) {
            system("clear");
            fprintf(output,"Вы ввели неправильное число, дозволено тільки ціле
число від %i до %i\n",min,max);
        }
    }
}
```

```

    }
    return number;
}

typedef struct element element;
struct element
{
    int value;
    struct element * next;
};

element * add_to_end(element * head, int value) {
    element * node = (element*) malloc(sizeof(element));
    node->value = value;
    node->next = NULL;
    if (head == NULL) {
        return node;
    }
    element * iter = head;
    while (iter->next != NULL) {
        iter=iter->next;
    }
    iter->next = node;
    return head;
}

element * fill_list(FILE * input, element * head, int count) {
    element * h = head, * node = head, * prev = NULL;
    int a = 0;
    while (node != NULL) {
        prev = node;
        node = node->next;
    }
    for (int i=0; (count==0 || i<count) && !feof(input); i++) {
        fscanf(input,"%i",&a);
        element * node = (element*) malloc(sizeof(element));
        node->value = a;
        node->next = NULL;
        if (prev == NULL)
            h = node;
        else
            prev->next = node;
        prev = node;
    }
    return h;
}

char show_list (FILE * output, element * head) {
    if (head == NULL) {
        fprintf(output,"Список пустий\n");
        return 1;
    }
    element * iter = head;
    while (iter->next != NULL) {
        printf("%i ",iter->value);
        iter=iter->next;
    }
    printf("%i\n",iter->value);
}

```

```

        return 0;
    }

size_t count_list (element * head) {
    if (head == NULL)
        return 0;
    size_t x=1;
    element * iter = head;
    while (iter->next != NULL) {
        x++;
        iter=iter->next;
    }
    return x;
}

element * find_by_key (element * head, int key) {
    if (head == NULL)
        return NULL;
    element * node = head;
    for (int i = 1; i < key; i++) {
        if (node->next == NULL) return NULL;
        node = node->next;
    }
    return node;
}

int find_by_value (element * head, int value) {
    if (head == NULL)
        return -1;
    element * node = head;
    for (int i = 1; node != NULL; i++) {
        if (node->value == value) return i;
        node = node->next;
    }
    return -1;
}

element * delete_node_by_value (element * head, int value) {
    if (head == NULL)
        return NULL;
    element * node = head, * prev = NULL, * next = NULL;
    while (node != NULL) {
        if (node->value == value) {
            next = node->next;
            free(node);
            if (prev != NULL) {
                prev->next = next;
                return head;
            } else return next;
        }
        prev = node;
        node = node->next;
    }
    return head;
}

element * delete_node_by_key (element * head, int key) {
    if (head == NULL)

```

```

        return NULL;
    element * node = head, * prev = NULL, * next = NULL;
    for (int i = 1; i < key; i++) {
        if (node == NULL) return head;
        prev = node;
        node = node->next;
    }
    next = node->next;
    free(node);
    if (prev != NULL) {
        prev->next = next;
        return head;
    } else return next;
}

```

```

void delete_even (element * head) {
    if (head == NULL)
        return;
    int len = count_list(head)/2;
    for (int i = 1; i < len; i++) {
        delete_node_by_key(head,i+1);
    }
}

```

```

/*
* function delete_even (list)
*   if list = null then
*       return
*   end if
*   len = count_list(head) / 2;
*   for i=1 to len step 1 do
*       delete_node_by_key(list,i+1)
*   end for
* end function
*
* Difficulty:
*  $T(\text{list}[1..n]) = 2 + (4+3*n) + 3*(n/2-1) + 9*(n/2-1) + 5*(2+3+...+n/2) =$ 
*  $= 9*n-6 + 5*(2+n/2)*(n/2-1)/2 =$ 
*  $= n^2/6+55/4*n-19/4=\theta(n^2)$ 
*  $M(\text{list}[1..n]) = 1+2+4 = 6 = \theta(1)$ 
* Funcion is quadratic
*/

```

```

void delete_even_fast (element * head) {
    if (head == NULL)
        return;
    element * node = head, * prev = NULL, * next = NULL;
    for (int i = 1; node != NULL; i=(i+1)%2) {
        next = node->next;
        if (i%2 == 0) {
            free(node);
            if (prev != NULL) {
                prev->next = next;
            }
        }
        prev = node;
        node = next;
    }
}

```



```

}

/*
* function delete_even_fast (list)
*   if list = null then
*     return
*   end if
*   node = list
*   i = 1
*   while node != null do
*     i = (i+1) % 2
*     next = node.next
*     if i % 2 = 0 then
*       FREE MEMORY (node)
*       if prev != null then
*         prev.next = next
*       end if
*     end if
*     prev = node
*     node = next
*   end while
* end function
*/

* Difficulty:  $4n/2 + 7n/2 + 3 = 11n/2 + 3$ 
*  $11n/2 + 3 = O(n)$ 
* Function is linear
*/

element * delete_list (element * head) {
    if (head == NULL)
        return NULL;
    element * iter = head;
    element * next;
    while (iter != NULL) {
        next = iter->next;
        free(iter);
        iter = next;
    }
    return NULL;
}

int main() {
    element * head = NULL, * node = NULL;
    int answer = 0, x = 0;
    FILE * input = NULL;
    while (answer != 12) {
        printf("Виберіть дію:\n");
        printf("1. Додати n елементів в кінець списку з клавіатури\n");
        printf("2. Додати n елементів в кінець списку з файлу data.txt\n");
        printf("3. Додати один елемент в кінець списку\n");
        printf("4. Видалити один елемент зі списку за значенням\n");
        printf("5. Видалити один елемент зі списку за номером\n");
        printf("6. Пошук елемента в списку за номером\n");
        printf("7. Пошук елемента в списку за значенням\n");
        printf("8. Вивести список на екран\n");
        printf("9. Вивести на екран поточну кількість елементів в списку\n");
        printf("10. Очищення списку\n");
        printf("11. Видалити елементи з парними номерами\n");
    }
}

```

```

printf("12. Вихід\n");
scanf("%i",&answer);
ign_other(stdin);
system("clear");
switch (answer) {
    case 1:
        head = fill_list(stdin, head,
get_number(stdin,stdout,"Введіть кількість",1,INT_MAX-1));
        ign_other(stdin);
        break;
    case 2:
        input = fopen("data.txt", "r");
        if (input == NULL) {
            printf("Такого файла не існує.\n");
            break;
        }
        head = fill_list(input, head,
get_number(stdin,stdout,"Введіть кількість",1,INT_MAX-1));
        break;
    case 3:
        head = add_to_end(head, get_number(stdin,stdout,"Введіть
число",INT_MIN+1,INT_MAX-1));
        break;
    case 4:
        show_list(stdout,head);
        head = delete_node_by_value(head,
get_number(stdin,stdout,"Введіть число",INT_MIN+1,INT_MAX-1));
        break;
    case 5:
        show_list(stdout,head);
        head = delete_node_by_key(head,
get_number(stdin,stdout,"Введіть номер",1,INT_MAX-1));
        break;
    case 6:
        node = find_by_key(head, get_number(stdin,stdout,"Введіть
номер",1,INT_MAX-1));
        if (node == NULL)
            printf("Елемент не знайдений\n");
        else
            printf("Елемент має значення: %i\n",(int)node-
>value);
        break;
    case 7:
        show_list(stdout,head);
        x = find_by_value(head, get_number(stdin,stdout,"Введіть
значення",INT_MIN+1,INT_MAX-1));
        if (x == -1)
            printf("Елемент не знайдений\n");
        else
            printf("Елемент знаходиться на %i позиції\n",x);
        break;
    case 8:
        break;
    case 9:
        printf("Список має таку кількість елементів: %lu\n",
count_list(head));
        break;
    case 10:

```

```

        head = delete_list(head);
        break;
    case 11:
        show_list(stdout, head);
        delete_even(head);
        break;
    case 12:
        break;
    default:
        printf("Неправильне значення, треба ввести число від 1 до
12\n");
    }
    if (answer != 12) {
        show_list(stdout, head);
        printf("Натисніть ENTER для продовження\n");
        ign_other(stdin);
        system("clear");
    }
}
delete_list(head);
return 0;
}

```

#### 4. Результати роботи програми:

##### Приклад 1:

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

2

Введіть кількість:

30

-1 -51 -78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8  
-70 -99 94 21 -100 44 -40

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку

11. Видалити елементи з парними номерами

12. Вихід

1

Введіть кількість:

5

1 2 3 4 5

-1 -51 -78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8  
-70 -99 94 21 -100 44 -40 1 2 3 4 5

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури

2. Додати n елементів в кінець списку з файлу data.txt

3. Додати один елемент в кінець списку

4. Видалити один елемент зі списку за значенням

5. Видалити один елемент зі списку за номером

6. Пошук елементу в списку за номером

7. Пошук елементу в списку за значенням

8. Вивести список на екран

9. Вивести на екран поточну кількість елементів в списку

10. Очищення списку

11. Видалити елементи з парними номерами

12. Вихід

3

Введіть число:

20

-1 -51 -78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8  
-70 -99 94 21 -100 44 -40 1 2 3 4 5 20

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури

2. Додати n елементів в кінець списку з файлу data.txt

3. Додати один елемент в кінець списку

4. Видалити один елемент зі списку за значенням

5. Видалити один елемент зі списку за номером

6. Пошук елементу в списку за номером

7. Пошук елементу в списку за значенням

8. Вивести список на екран

9. Вивести на екран поточну кількість елементів в списку

10. Очищення списку

11. Видалити елементи з парними номерами

12. Вихід

4

-1 -51 -78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8  
-70 -99 94 21 -100 44 -40 1 2 3 4 5 20

Введіть число:

-51

-1 -78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70  
-99 94 21 -100 44 -40 1 2 3 4 5 20

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури

2. Додати n елементів в кінець списку з файлу data.txt

3. Додати один елемент в кінець списку

4. Видалити один елемент зі списку за значенням

5. Видалити один елемент зі списку за номером

6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

4

-1 -78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70  
-99 94 21 -100 44 -40 1 2 3 4 5 20

Введіть число:

-1

-78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99  
94 21 -100 44 -40 1 2 3 4 5 20

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

4

-78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99  
94 21 -100 44 -40 1 2 3 4 5 20

Введіть число:

20

-78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99  
94 21 -100 44 -40 1 2 3 4 5

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

5

-78 61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99  
94 21 -100 44 -40 1 2 3 4 5

Введіть номер:

1

61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94  
21 -100 44 -40 1 2 3 4 5

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

5

61 49 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94  
21 -100 44 -40 1 2 3 4 5

Введіть номер:

2

61 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21  
-100 44 -40 1 2 3 4 5

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

5

61 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21  
-100 44 -40 1 2 3 4 5

Введіть номер:

31

61 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21  
-100 44 -40 1 2 3 4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами

12. Вихід

5

61 -69 -42 -65 73 -4 -89 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21  
-100 44 -40 1 2 3 4

Введіть номер:

7

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

6

Введіть номер:

1

Елемент має значення: 61

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

7

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

Введіть значення:

-42

Елемент знаходиться на 3 позиції

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку

4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елемента в списку за номером
7. Пошук елемента в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

8

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елемента в списку за номером
7. Пошук елемента в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

9

Список має таку кількість елементів: 29

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елемента в списку за номером
7. Пошук елемента в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

11

61 -69 -42 -65 73 -4 -98 79 -63 -22 -22 3 -80 -26 -31 1 -54 -8 -70 -99 94 21 -100  
44 -40 1 2 3 4

61 -42 73 -98 -63 -22 -80 -31 -54 -70 94 -100 -40 2 3 4

Натисніть ENTER для продовження

10

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером



6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

2

Введіть кількість:

10

61 -42 73 -98 -63 -22 -80 -31 -54 -70 94 -100 -40 2 3 4 -1 -51 -78 61 49 -69 -42  
-65 73 -4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

10

Список пустий

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

2

Введіть кількість:

10

-1 -51 -78 61 49 -69 -42 -65 73 -4

Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елементу в списку за номером
7. Пошук елементу в списку за значенням

8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

11  
-1 -51 -78 61 49 -69 -42 -65 73 -4  
-1 -78 49 -42 73 -4  
Натисніть ENTER для продовження

Виберіть дію:

1. Додати n елементів в кінець списку з клавіатури
2. Додати n елементів в кінець списку з файлу data.txt
3. Додати один елемент в кінець списку
4. Видалити один елемент зі списку за значенням
5. Видалити один елемент зі списку за номером
6. Пошук елемента в списку за номером
7. Пошук елемента в списку за значенням
8. Вивести список на екран
9. Вивести на екран поточну кількість елементів в списку
10. Очищення списку
11. Видалити елементи з парними номерами
12. Вихід

12

5. Псевдокод функції з пункту b:

```
function delete_even (list)
  if list = null then
    return
  end if
  len = count_list(head) / 2;
  for i=1 to len step 1 do
    delete_node_by_key(list,i+1)
  end for
end function
```

Так як для count\_list  $T(n)=4+3n$  , а для delete\_node\_by\_key:

$$\begin{cases} T(n,i)=9+5i & , \text{якщо } i \leq n \\ T(n,i)=9+5n & , \text{якщо } i > n \end{cases}$$

А так як delete\_even має 2 операції за циклом та 3 операції у циклі, а у циклі n-1 ітерацій, то з цих формул маємо:

$$T(n)=2+(4+3*n)+3*\left(\frac{n}{2}-1\right)+9*\left(\frac{n}{2}-1\right)+5*(2+3+\dots+\frac{n}{2})$$

$$T(n)=9*n-6+5*(2+\frac{n}{2})*(\frac{n}{2}-1)/2=\frac{n^2}{6}+\frac{55*n}{4}-\frac{19}{4}=\Theta(n^2)$$

```
function delete_even_fast (list)
  if list = null then
    return
  end if
  node = list
  i = 1
  while node != null do
```

```

        i = (i+1) % 2
        next = node.next
        if i % 2 = 0 then
            FREE MEMORY (node)
            if prev != null then
                prev.next = next
            end if
        end if
        prev = node
        node = next
    end while
end function

```

Код за циклом має 3 операції, а у циклі для парних елементів 5 операцій, а для непарних 8 операцій, тобто загальна кількість операцій:

$$T(n) = \frac{5*n}{2} + \frac{8*n}{2} + 3 = \frac{13*n}{2} + 3$$

З формули можна припустити, що функція має лінійну швидкість, перевіримо це:

$$\lim_{n \rightarrow \infty} \frac{\frac{13*n}{2} + 3}{n} = 13$$

Що означає що:

$$T(n) = \frac{13*n}{2} + 3 = \Theta(n)$$

Це означає, що функція має лінійну швидкість.