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:
   1. 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.

* + 1. 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.

* + 1. Пошук елементу в списку за номером.

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

* + 1. 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.

* + 1. Виведення списку на екран.
    2. Виведення на екран поточної кількості елементів в списку.
    3. 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.

* 1. Реалізувати дії згідно з варіантом, використовуючи розроблені операції зі списком.
     1. Реалізувати алгоритм, який видаляє кожний другий елемент списку.

1. Опис структури программи та реалізованих функцій:
   1. 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
   2. Программа 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 программа забере тільки число.
         2. Аргументи:
            1. input - вхідний поток.
         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
   3. 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
   4. 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
2. 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 неправильне 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(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=Θ(n^2)

\* M(list[1..n]] = 1+2+4 = 6 = Θ(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: 4\*n/2+7\*n/2+3 = 11\*n/2+3

\* 11\*n/2+3=Θ(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("Вибeріть 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 елементу в списку за номером\n");

printf("7. Пошук елементу 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 кількіть",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;

}

1. Результати роботи програми:

Приклад 1:

Вибeріть 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

Вибeріть 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 кількіть:

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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

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

Вибeріть 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

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

Вибeріть 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

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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

Вибeріть 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

12

1. 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 , а для delete\_node\_by\_key:

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:

**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:

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

Що означає що:

Це означає, що функція має лінійну швидкість.