

## ЛАБОРАТОРНА РОБОТА № 2.1.

### РЕКУРСИВНІ АЛГОРИТМИ

#### Мета лабораторної роботи

Метою лабораторної роботи № 2.1 є засвоєння теоретичного матеріалу та набуття практичного досвіду створення рекурсивних алгоритмів та написання відповідних їм програм.

#### Постановка задачі

Дане натуральне число  $n$ . Знайти суму перших  $n$  членів ряду чисел, заданого рекурентною формулою. Розв'язати задачу трьома способами (написати три програми): 1) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску; 2) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні; 3) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні.

Програми повинні працювати коректно для довільного натурального  $n$  включно з  $n = 1$ .

#### Зміст звіту

1. Загальна постановка задачі та завдання для конкретного варіанту.
2. Текст усіх трьох програм.
3. Тестування програм. З метою тестування потрібно написати циклічний варіант рішення задачі, а також виконати обчислення заданої формули на калькуляторі.
4. Як результат, роздрукувати дані тестування (для  $n=5$ ) як циклічною програмою, так і обчислення на калькуляторі, та розв'язку задачі усіма трьома рекурсивними програмами. Результати обчислень усіма способами повинні співпадати.

5. Алгоритм кожного варіанту обчислює відповідну елементарну функцію з деякою похибкою. Графік цієї похибки, залежно від вхідних даних, також повинна бути представлена у звіті.

### Контрольні питання

1. Визначення рекурсивного об'єкту.
2. Визначення глибини та поточного рівня рекурсії.
3. Форма виконання рекурсивних дій на рекурсивному спуску.
4. Форма виконання рекурсивних дій на рекурсивному поверненні.
5. Форма виконання рекурсивних дій на як рекурсивному спуску, так і на рекурсивному поверненні.

### Варіанти індивідуальних завдань

#### Варіант № 1

$$F_1 = 1; F_2 = -x/2; F_i = -F_{i-1} \cdot x \cdot (2i - 3)/(2i), i > 2;$$

$$\sum F_i = \sqrt{1-x}, |x| < 1.$$

#### Варіант № 2

$$F_1 = 1; F_2 = -x/3; F_i = -F_{i-1} \cdot x \cdot (3i - 7)/(3i - 3), i > 2;$$

$$\sum F_i = \sqrt[3]{1-x}, |x| < 1.$$

#### Варіант № 3

$$F_1 = 1; F_i = -F_{i-1} \cdot x \cdot (2i - 3)/(2i - 2), i > 1;$$

$$\sum F_i = 1/\sqrt{1+x}, |x| < 1.$$

#### Варіант № 4

$$F_1 = 1; F_i = -F_{i-1} \cdot x \cdot (3i - 5)/(3i - 3), i > 1;$$

$$\sum F_i = 1/\sqrt[3]{1+x}, |x| < 1.$$

#### Варіант № 5

$$F_1 = 1; F_i = F_{i-1} \cdot (2x/3 - 1); i > 1;$$

$$\sum F_i = 1.5/x, 1 < x < 2.$$

Вариант № 6

$$F_1 = 1; \quad F_i = F_{i-1} \cdot x \cdot \ln 2 / i; \quad i > 1;$$

$$\sum F_i = 2^x.$$

Вариант № 7

$$F_1 = (x - 1) / (x + 1); \quad F_{i+1} = F_i \cdot (2i - 1)(x - 1)^2 / ((2i + 1) \cdot (x + 1)^2); \quad i > 1;$$

$$\sum F_i = -0.5 \ln x; \quad x < 1.$$

Вариант № 8

$$F_1 = x - 1; \quad F_{i+1} = -F_i \cdot (x - 1) \cdot i / (i + 1); \quad i > 0;$$

$$\sum F_i = \ln x, \quad 0 < x < 2;$$

Вариант № 9

$$F_1 = (x - 1) / x; \quad F_{i+1} = F_i \cdot i \cdot (x - 1) / (i \cdot x + x); \quad i > 0;$$

$$\sum F_i = \ln x; \quad x > 0.5.$$

Вариант № 10

$$F_1 = x; \quad F_2 = -x^2 / 2; \quad F_i = -F_{i-1} \cdot x \cdot (i - 1) \cdot (i - 2) / (i^2 - i); \quad i > 2;$$

$$(1+x) \sum F_i = \ln(1+x); \quad -1 < x < 2.$$

Вариант № 11

$$F_1 = x; \quad F_{i+1} = -F_i \cdot x^2 / (4i^2 + 2i); \quad i > 0;$$

$$\sum F_i = \sin x.$$

Вариант № 12

$$F_1 = 1; \quad F_{i+1} = -F_i \cdot x^2 / (4i^2 - 2i); \quad i > 0;$$

$$\sum F_i = \cos x.$$

Вариант № 13

$$F_1 = x; \quad F_{i+1} = F_i \cdot x^2 (2i - 1)^2 / (4i^2 + 2i); \quad i > 0;$$

$$\sum F_i = \arcsin x, \quad -1 < x < 1.$$

Вариант № 14

$$F_0 = x; \quad F_i = F_{i-1} \cdot (2i - 1)^2 \cdot x^2 / (2i(2i + 1)); \quad i > 0;$$

$$\pi/2 - \sum F_i = \arccos x.$$

Вариант № 15

$$F_1 = x; F_{i+1} = -F_i \cdot x^2 (2i-1)/(2i+1); i > 0;$$

$$\sum F_i = \operatorname{arctg} x, |x| < 1;$$

Вариант № 16

$$F_1 = x; F_{i+1} = F_i \cdot (2i-1)^2 \cdot x^2/(4i^2 + 2i); i > 0;$$

$$\sum F_i = \operatorname{arsin} x, |x| < 1;$$

Вариант № 17

$$F_1 = x; F_{i+1} = F_i \cdot x^2/(4i^2 + 2i); i > 0;$$

$$\sum F_i = \operatorname{sh} x, |x| < 10^6;$$

Вариант № 18

$$F_1 = 1; F_{i+1} = F_i \cdot x^2/(4i^2 - 2i); i > 0;$$

$$\sum F_i = \operatorname{ch} x, |x| < 10^6;$$

Вариант № 19

$$F_1 = x; F_{i+1} = -F_i \cdot x^2 (2i-1)^2/(4i^2 + 2i); i > 0;$$

$$\sum F_i = \operatorname{arcsch} x, |x| < 1;$$

Вариант № 20

$$F_1 = x; F_{i+1} = F_i \cdot x^2(2i-1)/(2i+1); i > 0;$$

$$\sum F_i = \operatorname{arcth} x, |x| < 1;$$

Вариант № 21

$$F_0 = 4/3; F_1 = x/2; F_{i+1} = -F_i \cdot x (2i-1)/2i; i > 1;$$

$$\sum F_i = \sqrt{1+x}, |x| < 1;$$

Вариант № 22

$$F_1 = x-1; F_{i+1} = -F_i \cdot (i-1)(x-1)/i; i > 0;$$

$$\sum F_i = \ln x, 0 < x < 2;$$

Вариант № 23

$$F_1 = (x-1)/x; F_i = F_{i-1} \cdot (i-1)(x-1)/(ix); i > 1;$$

$$\sum F_i = \ln x, 0.5 < x;$$

Вариант № 24

$$F_0 = 1; F_{i+1} = -F_i \cdot x^2/i; i > 0;$$

$$\sum F_i = e^{-x};$$

Вариант № 25

$$F_1 = x; F_i = -F_{i-1} \cdot x(i-1)/i; i > 1;$$

$$\sum F_i = \ln(1+x), -1 < x < 1;$$

Вариант № 26

$$F_1 = x; F_i = F_{i-1} \cdot x^2(2i-3)/(2i-1); i > 1;$$

$$\sum F_i = 0.5 \ln((1+x)/(1-x)), -1 < x < 1;$$

Вариант № 27

$$F_1 = 4/3; F_{i+1} = F_i \cdot (1 - 4x/3); i > 0;$$

$$\sum F_i = 1/x, 0.5 < |x| < 1;$$

Вариант № 28

$$F_1 = 1; F_{i+1} = -F_i \cdot x^2/(4i^2 - 2i); i > 0;$$

$$\sum F_i = \operatorname{ch} x.$$

Вариант № 29

$$F_1 = 0.5901(x + 0.7071); z = x / F_1^2 - 1; F_{i+1} = F_i \cdot z(3 - 2i)/(2i); i > 0;$$

$$\sum F_i = \sqrt{x}, 0.5 < x < 1.$$

Вариант № 30

$$F_1 = 0.4153(x + 1.4237); z = x / F_1^3 - 1; F_{i+1} = F_i \cdot z(4 - 3i)/(3i); i > 0;$$

$$\sum F_i = \sqrt[3]{x}, 0.5 < x < 1.$$

Вариант № 31

$$F_1 = x; F_{i+1} = F_i \cdot x^2/(4i^2 + 2i); i > 0;$$

$$\sum F_i = \operatorname{sh} x.$$

Вариант № 32

$$F_1 = 1.951 - x; F_{i+1} = 0.5 F_i \cdot (1 - x \cdot F_1^2)(1 + i)/i; i > 0;$$

$$\sum F_i = 1/\sqrt{x}, 0.5 < x < 1.$$

Вариант № 33

$$F_1 = 1; \quad F_i = F_{i-1} \cdot x \cdot \ln 2 / i; \quad i > 1;$$

$$\sum F_i = 2^x.$$

## ЛАБОРАТОРНА РОБОТА №2.2.

### ЗВ'ЯЗАНІ ДИНАМІЧНІ СТРУКТУРИ ДАНИХ.

#### СПИСКИ

#### Мета лабораторної роботи

Метою лабораторної роботи №2.2 є засвоєння теоретичного матеріалу та набуття практичного досвіду використання зв'язаних динамічних структур даних у вигляді одно- та двозв'язних списків при складанні різних алгоритмів.

#### 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  $2n$ ) **невідомо** 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).

### **Зміст звіту**

1. Загальна постановка задачі та завдання для конкретного варіанту.
2. Текст програми.
3. Тестування програми, тобто початкові дані та відповідні їм результати.
4. В якості результату роздрукувати дані тестування та розв'язку задачі на комп'ютері.

### **Контрольні питання**

1. Чи існують обмеження на кількість елементів у списку, якщо так, то які?
2. Які є види спискових структур даних з точки зору їх логічного використання (стек, черга, тощо). Поясніть їх особливості та відмінності між ними.
3. Які є вимоги до структури елемента зв'язних динамічних даних?
4. В чому полягає особливість опису типів для створення зв'язних динамічних даних?
5. Якою повинна бути структура елемента лінійного двозв'язного списку?
6. Скільки вказівників і якого призначення необхідно для роботи з чергою?
7. Скільки вказівників і якого призначення необхідно для роботи зі стеком?
8. Скільки вказівників і якого призначення необхідно для роботи з деком?
9. Скільки вказівників і якого призначення необхідно для роботи з лінійними однозв'язними списками?
10. Скільки вказівників і якого призначення необхідно для роботи з лінійними двозв'язними списками?



## Варіанти завдань

### Варіант 1

Ключами елементів списку є рядки довжиною не більше 10-ти символів, що складаються з латинських літер. Відсортувати елементи списку у лексикографічному порядку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»), методом вибору.

### Варіант 2

Ключами елементів списку є дійсні числа. Кількість елементів списку повинна дорівнювати  $2n$ . Обчислити значення виразу:  $(a_1 - a_{2n})(a_3 - a_{2n-2}) \dots (a_{2n-1} - a_2)$ , де  $a_i$  –  $i$ -й елемент списку.

### Варіант 3

Ключами елементів списку є символи з множини латинських літер та цифр. Перекомпонувати список таким чином, щоб усі цифри стояли на початку списку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 4

Заданий список є *чергою*. Ключами елементів списку є символи. Переписати елементи цієї черги до іншої черги у оберненому порядку, *не підраховуючи кількості елементів у черзі*.

### Варіант 5

Ключами елементів списку є дійсні числа. Виконати циклічний зсув елементів списку на  $k$  позицій вліво ( $k$  – натуральне і не перевищує кількості елементів списку). При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

#### Варіант 6

Ключами елементів списку є дійсні числа. Відсортувати елементи списку за незбільшенням, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»), методом обміну (“бульбашки”) з використанням «прапорця».

#### Варіант 7

Ключами елементів списку є цілі числа. Кількість елементів списку повинна дорівнювати  $2n$ . Обчислити значення виразу:  $a_1 a_{2n} + a_2 a_{2n-1} + \dots + a_n a_{n+1}$ , де  $a_i$  –  $i$ -й елемент списку.

#### Варіант 8

Ключами елементів списку є цілі ненульові числа, причому кількість від’ємних чисел дорівнює кількості додатних. Перекомпонувати список так, щоб отримати послідовність чисел із чергуванням знаків, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

#### Варіант 9

Ключами елементів списку є різні дійсні числа. Знайти максимальний та мінімальний елементи списку. Вставити до списку два нових елементи: після елемента з максимальним значенням вставити елемент з мінімальним значенням, а після елемента з мінімальним значенням вставити елемент з максимальним значенням.

#### Варіант 10

Ключами елементів списку є цілі числа. Визначити кількість елементів списку, значення яких більше за задане ціле число  $M$ , та вставити нових елементів після  $k$ -го ( $k$  – натуральне і не перевищує кількості елементів списку) елемента списку.

### Варіант 11

Ключами елементів списку є латинські літери. Відсортувати елементи списку у лексикографічному порядку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»), методом вставки.

### Варіант 12

Ключами елементів списку є цілі числа. Обчислити значення виразу:  $(a_1 + a_2 + 2a_n)(a_2 + a_3 + 2a_{n-1}) \dots (a_{n-1} + a_n + 2a_2)$ , де  $a_i$  –  $i$ -й елемент списку.

### Варіант 13

Ключами елементів списку є цілі числа. Переставити елементи списку так, щоб спочатку розташовувались додатні, потім нульові, а за ними від'ємні елементи, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 14

Ключами елементів списку є рядки довжиною не більше 5-ти символів. Перекомпонувати список так, щоб елементи списку були розташовані у оберненому порядку (виконати «дзеркальне відображення» списку), не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 15

Ключами елементів списку є дійсні числа. Перекомпонувати елементи списку таким чином, щоб його елементи розташовувались у такому порядку:  $a_1, a_n, a_2, a_{n-1}, \dots, a_{\lceil \frac{n+1}{2} \rceil}$ , де  $a_i$  –  $i$ -й елемент списку. При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

#### 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»).

#### Варіант 17

Ключами елементів списку є дійсні числа. Обчислити значення виразу:  $a_1 a_n + a_2 a_{n-1} + \dots + a_n a_1$ , де  $a_i$  –  $i$ -й елемент списку.

#### Варіант 18

Ключами елементів списку є цілі ненульові числа, які розташовуються у наступному порядку: 10 додатних, 10 від'ємних і т. д. Кількість елементів списку  $n$  повинна бути кратною 20. Перекомпонувати список так, щоб розташування елементів було наступним: 5 додатних, 5 від'ємних і т. д., не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

#### 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»).

#### Варіант 20

Ключами елементів списку є цілі числа. Кількість елементів списку повинна дорівнювати  $2n$ . Перекомпонувати елементи списку так, розташування елементів було наступним:  $a_1, a_{n+1}, a_2, a_{n+2}, a_3, \dots, a_n, a_{2n}$ , де  $a_i$  –  $i$ -й компонент списку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 21

Ключами елементів списку є дійсні числа. Виконати наступні дії: якщо елементи списку впорядковані за незбільшенням, то залишити його без змін, інакше перезаписати елементи списку у оберненому порядку. При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

### Варіант 22

Ключами елементів списку є цілі ненульові числа, які розташовуються в наступному порядку: 5 від'ємних, 5 додатних і т. д. Кількість елементів списку  $n$  повинна бути кратною 20-ти. Перекомпонувати елементи списку так, щоб розташування елементів було наступним: 10 від'ємних, 10 додатних і т. д., не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 23

Ключами елементів списку є цілі числа. Виконати циклічний зсув елементів списку на  $k$  позицій вправо ( $k$  – натуральне і не перевищує кількості елементів списку). При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

### Варіант 24

Ключами елементів списку є цілі ненульові числа. Кількість елементів списку  $n$  повинна бути кратною 4-ом, причому кількість від'ємних чисел дорівнює кількості додатних. Перекомпонувати елементи списку так, розташування елементів було наступним: два додатних, два від'ємних і т. д., не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 25

Ключами елементів списку є латинські літери. Перекомпонувати список так, щоб спочатку розташовувались елементи з голосними латинськими літерами, а потім елементи з приголосними латинськими літерами, не змінюючи початкового взаємного розташування літер. Наприклад:

початковий список: university

результат: uieiyinvrst

При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

### Варіант 26

Ключами елементів списку є натуральні числа. Перекомпонувати список так, щоб спочатку йшли елементи з ключами, що діляться на 3 без залишку, потім елементи з ключами, що діляться на 3 із залишком 1, і нарешті ті, що діляться на 3 із залишком 2, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

### Варіант 27

Ключами елементів списку є цілі числа. Перекомпонувати список так, щоб спочатку розташовувались додатні, потім нульові, а за ними від'ємні елементи, не змінюючи початкового взаємного розташування елементів. Наприклад:

початковий файл: -1 0 5 -9 8 -3 5 0 -7 4

результат: 5 8 5 4 0 0 -1 -9 -3 -7.

При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

### 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»).

### Варіант 29

Ключами елементів списку є рядки довжиною не більше 25-ти символів. Кількість елементів списку  $n$  повинна бути кратною 20-ти. Перекомпонувати список всередині кожних 20-ти елементів, розташувавши їх у наступному порядку:  $a_1, a_{11}, a_2, a_{12}, \dots, a_{21}, a_{31}, a_{22}, a_{32}, \dots$ , де  $a_i$  –  $i$ -й елемент списку. При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

### Варіант 30

Ключами елементів списку є цілі числа. Відсортувати елементи списку за незменшенням, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»), методом шейкерного сортування.