Побудова алгоритмів складних виразів

к.т.н., доцент кафедри прикладної математики
Рижа Ірина Андріївна

Про що ця лекція???

- Опишемо поняття про структурований підхід при побудові алгоритмів.
- Викладемо особливості алгоритмів складних математичних виразів.
- Розглянемо задачі обробки однотипних наборів даних.

Структурований підхід при побудові алгоритмів

Структурований підхід

– це підхід, який

- дозволяє "дисциплінувати" процес створення (побудови) алгоритму;
- не є обов'язковим при складанні алгоритмів;
- його застосування значно полегшує як читання, так і аналіз алгоритму виконавцями.

Основні структуровані елементи

1. Слідування
2. Розгалуження
3. Обхід
4. Цикли

Основні структуровані елементи

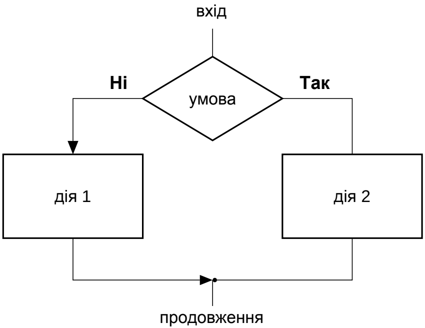
1. **Слідування**

   - Усі символи і блоки алгоритму повинні бути послідовно розміщеними.
   - Алгоритм повинен читатись зліва-направо і зверху-вниз.

Основні структуровані елементи

2. Розгалуження

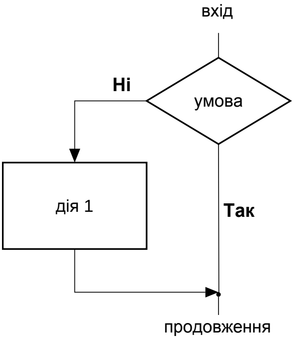
- Застосовується, коли в залежності від виконання умови (істинності умови) необхідно виконати одну або іншу дію.
- Кожна з цих дій, в свою чергу, може містити декілька етапів.



Основні структуровані елементи

3. Обхід

– частковий випадок розгалуження, коли одна з віток НЕ містить жодних дій.



Основні структуровані елементи

4. Цикли

– повторювана послідовність операцій.

- ***Тіло циклу*** – це сукупність етапів, які необхідно багаторазово повторювати в процесі виконання алгоритму.
- ***Умова завершення циклу*** – умова, яка задає число повторень тіла циклу.
- ***Ініціалізація циклу*** – присвоєння початкових значень ряду змінних, які входять в цикл, перед його виконанням.

Типи циклів

1. цикли з відомим числом повторень (цикли типу перерахунку);
2. цикли з невідомим числом повторень (ітераційні цикли).

Типи циклів

Цикли з відомим числом повторень *(післяумова)*

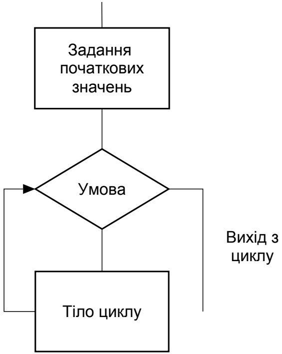
- число повторень тіла циклу **відоме**;
- лічильник числа повторень тіла циклу може змінювати свої значення лише в строго заданому інтервалі;
- перед виконанням групи операторів тіла циклу необхідно зарезервувати ряд змінних, які будуть зберігати **початкове** і **кінцеве** значення керуючої змінної, а також **величину кроку**, за допомогою якого ми від початкового значення керуючої змінної приходимо до її кінцевого значення.

Структура циклу з відомим числом повторень

Типи циклів

Цикли з невідомим числом повторень *(передумова)*

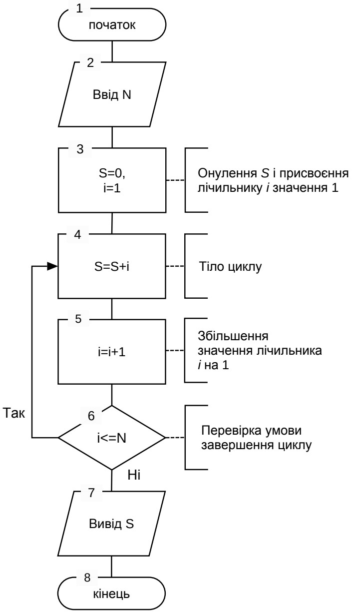
- умова виходу з циклу перевіряється перед виконанням операторів тіла циклу;
- в тілі циклу змінювати значення змінних, що входять в умову завершення циклу.

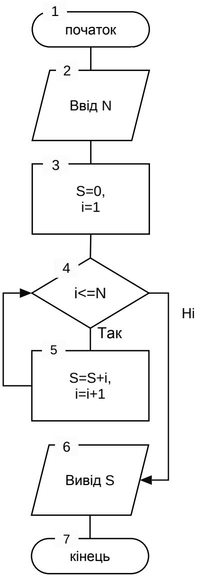


Приклад 1.

Обчислити суму перших N натуральних чисел.

- i – лічильник;
- S – шукана сума.


1
початок
2
Ввід N
3
S=0,
i=1
Онулення S і присвоєння лічильнику i значення 1
4
S=S+i
Тіло циклу
5
i=i+1
Збільшення значення лічильника i на 1
Так
6
i<=N
Перевірка умови завершення циклу
Ні
7
Вивід S
8
кінець


1
початок
2
Ввід N
3
S=0,
i=1
4
i<=N
Ні
Так
5
S=S+i,
i=i+1
6
Вивід S
7
кінець

Обчислення сум та добутків

Характерні ознаки:

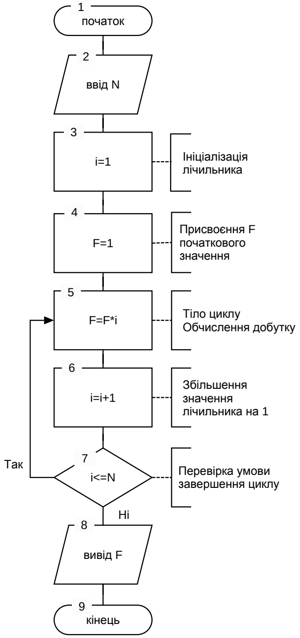
- змінній, якою позначаємо **суму**, необхідно надати початкового значення **0**;
- змінній, якою позначаємо **добуток**, необхідно надати початкового значення **1**.

Приклад 2.

Обчислити факторіал натурального числа N.

$$N! = 1{\cdot}2{\cdot}3{\cdot}\ldots{\cdot}(N-1){\cdot}N, \quad 0! = 1.$$

- i – лічильник;
- F – результат.


1
початок
2
ввід N
3
i=1
Ініціалізація лічильника
4
F=1
Присвоєння F початкового значення
5
F=F*i
Тіло циклу Обчислення добутку
6
i=i+1
Збільшення значення лічильника на 1
7
i<=N
Так
Ні
Перевірка умови завершення циклу
8
вивід F
9
кінець

Побудова алгоритмів складних виразів

У математиці сукупність елементів виду:

$$1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} \tag{1}$$

або

$$\frac{N+1}{1} \cdot \frac{N+2}{2} \cdot \frac{N+3}{3} \cdot \ldots \cdot \frac{N+k}{k} \tag{2}$$

називають **частинами рядів** і завдання обчислення відповідно суми або добутку позначають:

$$\sum_{i=0}^{n} \frac{x^i}{i!} \quad \text{або} \quad \prod_{i=1}^{k} \frac{N+i}{i}. \tag{3}$$

- З метою прискорення виконання алгоритму (шляхом зменшення числа операцій), доцільно встановити **залежність між наступним і попереднім членами ряду**, якщо це тільки можливо.

Приклад 3.

Побудувати алгоритм обчислення суми частини ряду:

$$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} = \sum_{i=0}^{n} \frac{x^i}{i!}.$$

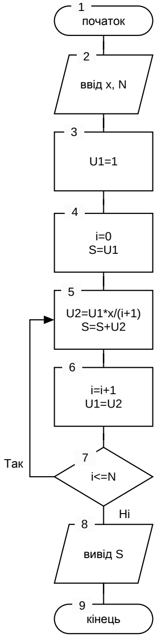
Сусідні члени ряду:

$$U_i = \frac{x^i}{i!}, \quad U_{i+1} = \frac{x^{i+1}}{(i+1)!}.$$

Тоді

$$\frac{U_{i+1}}{U_i} = \frac{\dfrac{x^{i+1}}{(i+1)!}}{\dfrac{x^i}{i!}} = \frac{x}{i+1} \Rightarrow U_{i+1} = \frac{x}{i+1} U_i.$$

- N – число членів ряду, які сумуються;
- $U1$, $U2$ – змінні, зарезервовані під U_i та U_{i+1};
- S – шукана сума ряду.


1
початок
2
ввід x, N
3
U1=1
4
i=0
S=U1
5
U2=U1*x/(i+1)
S=S+U2
6
i=i+1
U1=U2
7
i<=N
Так
Ні
8
вивід S
9
кінець

Одновимірні масиви

Однотипний набір даних –

сукупність даних, усі елементи якої належать до однієї множини.

Масив –

це впорядкована сукупність однотипних даних.

- для кожного елемента масиву можна вказати елемент, який йому передує і елемент, що слідує за ним.

Наприклад, тиждень (T) – це масив із семи елементів (днів):

Одновимірні масиви

Індексні змінні –

змінні величини, за допомогою яких можна звернутись до будь-якого елементу масиву.

$$i, \quad 1 \leqslant i \leqslant 7.$$

Індексована змінна –

змінна величина, яка в своєму позначенні містить індексну змінну.

$$T_i, \quad 1 \leqslant i \leqslant 7.$$

Одновимірний масив –

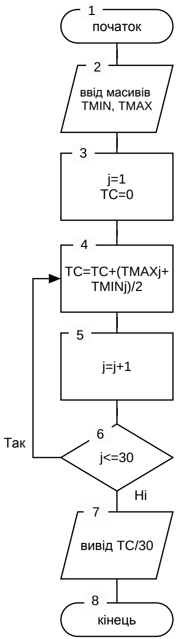
масив, для звертання до елементів якого достатньо однієї індексної змінної.

Приклад 4.

Робота з одновимірним масивом

Протягом квітня ми щоденно реєстрували максимальну і мінімальну температуру кожного дня і заносили отримані дані у дві таблиці $TMAX$ та $TMIN$. Побудувати алгоритм знаходження середньомісячної температури у квітні.

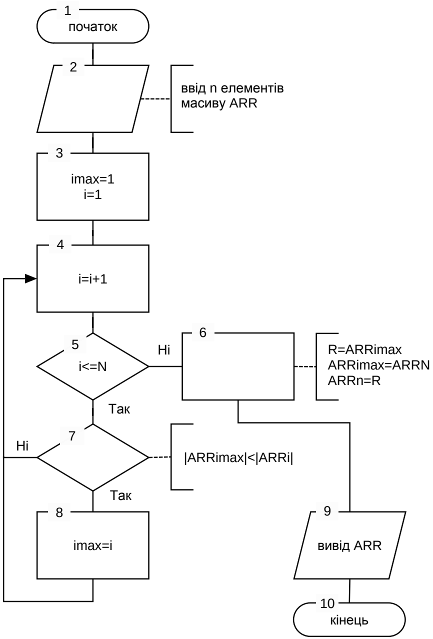
- $TMAX$, $TMIN$ – два одновимірні масиви, кожен з яких містить по 30 елементів;
- j, $1 \leqslant j \leqslant 30$ – індексна змінна;
- TC – результат.


1
початок
2
ввід масивів
TMIN, TMAX
3
j=1
TC=0
4
TC=TC+(TMAXj+
TMINj)/2
5
j=j+1
6
j<=30
Так
Ні
7
вивід TC/30
8
кінець

Приклад 5.

Знайти найбільший за модулем елемент у масиві з n дійсних чисел та обміняти його з останнім елементом масиву.

- ARR – одновимірний масив дійсних чисел;
- i, $1 \leqslant i \leqslant n$ – індексна (цілочисельна) змінна;
- $imax$ – номер найбільшого за модулем елемента масиву ARR;
- R – додаткова змінна.


1
початок
2
ввід n елементів
масиву ARR
3
imax=1
i=1
4
i=i+1
5
i<=N
Ні
6
R=ARRimax
ARRimax=ARRN
ARRn=R
Так
7
Ні
|ARRimax|<|ARRi|
Так
8
imax=i
9
вивід ARR
10
кінець

Дякую за увагу!

Далі буде...