

АЛГОРИТМІЗАЦІЯ.
СХЕМИ
АЛГОРИТМІВ.
СТРУКТУРИ ДАНИХ



Алгоритм - це довільна, коректна обчислювальна процедура, на вхід якої подається деяка величина або набір величин, а результатом виконання якої є вихідна величина або набір значень

ПОНЯТТЯ АЛГОРИТМІВ

- ⦿ Алгоритм – це кінцева система команд, адресованих виконавцю, які чітко і однозначно задають процес вирішення завдання у всіх деталях
- ⦿ Алгоритм – деякий скінченний впорядкований набір дій, послідовне виконання яких через скінченну кількість кроків приводить до розв'язку задачі.

ПОНЯТТЯ АЛГОРИТМІВ

- Алгоритм - набір систематизованих правил виконання обчислювального процесу, що обов'язково приводить до розв'язання певного класу задач після скінченного числа операцій

Властивості алгоритмів

дискретність

визначеність

виконуваність

результативність

масовість

формальність

ВЛАСТИВОСТІ АЛГОРИТМІВ.

ДИСКРЕТНІСТЬ

- Дискретність – розчленованість процесу, що визначається алгоритмом, на окремі елементарні операції, можливість виконання яких не викликає сумнівів.
- Процес, який визначається алгоритмом, повинен мати дискретний характер, тобто являти собою послідовність окремих кроків.
- Кожна наступна команда починає виконуватися після завершення виконання поточної команди
- Кожний алгоритм має справу з дискретними даними: вхідними, проміжними, вихідними (конечні визначені дані)
- Алгоритм виконується по кроках та при цьому на кожному кроці виконується тільки одна операція

Знайти середнє арифметичне чисел
[3, 5, 7, 9, 11].

ВЛАСТИВОСТІ АЛГОРИТМІВ. ВИЗНАЧЕНІСТЬ

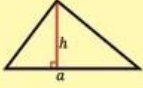
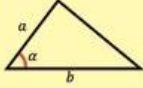
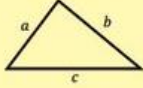
- Визначеність (детермінованість) (determinacy) – набір вказівок має бути точний, не залежати від виконавця.
- забезпечує однозначність результату процесу, що описується ним при заданих початкових даних.
- Кожен крок повинен бути чітко та недвозначно визначений і не допускати довільного трактування виконавцем



(рецепт - сіль за смаком)

ВЛАСТИВОСТІ АЛГОРИМІВ. МАСОВІСТЬ

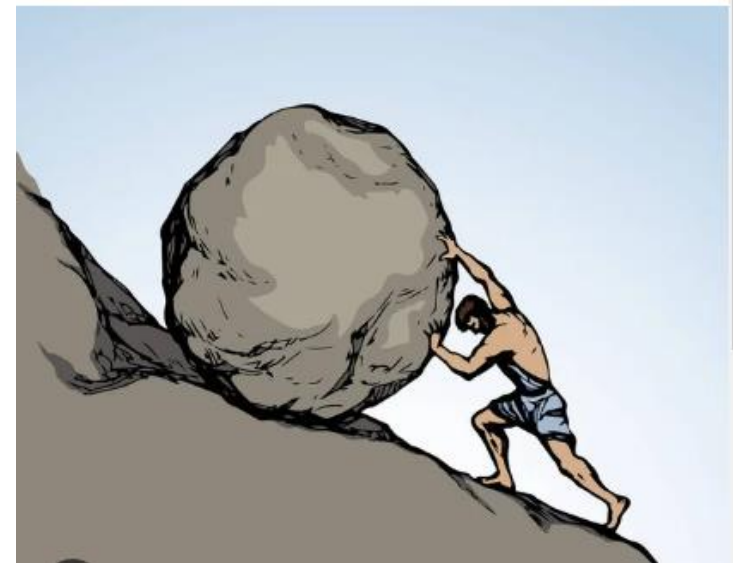
- Масовість — застосовність алгоритму до будь-яких даних задач певного класу.
- забезпечує розв'язання будь-якої задачі з класу однотипних задач при будь-яких початкових даних
- Наприклад, алгоритм обчислення площі трикутника застосовується до будь-яких трикутників.

| | | |
|---|---|---|
|  |  |  |
| $S = \frac{1}{2}ah$ | $S = \frac{1}{2}ab \sin \alpha$ | Формула Герона $S = \sqrt{p(p-a)(p-b)(p-c)},$ где $p = \frac{a+b+c}{2}$ |



ВЛАСТИВОСТІ АЛГОРИТМІВ. РЕЗУЛЬТАТИВНІСТЬ

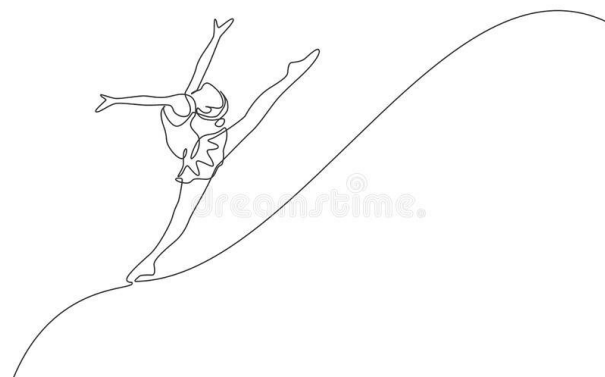
- Результативність — кінцівка процесу перетворення вхідної інформації у вихідну.
- Результативність вказує на те, що застосування алгоритму до будь-якого допустимого набору вхідних даних за кінцеве число кроків забезпечує отримання певного результату.
- Під час виконання алгоритму деякі його кроки можуть повторюватися багато разів, однак виконання алгоритму закінчиться за кінцеве число кроків.



ВИКОНУВАНІСТЬ

- Алгоритм містить команди, що можуть бути виконані виконавцем, він здатен їх виконати, вони входять до системи команд виконавця
- В алгоритмі не повинно бути невиконуваних операцій (наприклад, «поділ на нуль»)
- Чи можна записати алгоритм азбукою Морзе чи ієрогліфами?

| | | |
|--------|--------|---------|
| A ●- | J ●--- | S ●●● |
| B -●●● | K -●- | T - |
| C -●-● | L ●-●● | U ●●- |
| D -●● | M -- | V ●●●- |
| E ● | N -● | W ●-- |
| F ●●-● | O ---- | X -●●- |
| G --● | P ●-●● | Y -●-- |
| H ●●●● | Q --●- | Z ---●● |
| I ●● | R ●-● | |



ФОРМАЛЬНІСТЬ

- Один і той самий алгоритм може виконати будь-хто з однаковим результатом при однакових початкових даних



КЛАСИ АЛГОРИТМІВ



Обчислювальні алгоритми

- Алгоритми, які реалізують скінченну сукупність обчислювальних дій над простими видами даних



Інформаційні алгоритми

- Характерні для процедур із великими обсягами даних, що передбачають пошук, заміну, сортування



Управляючі алгоритми

- Являють собою набір вказівок, які залежать від даних, що надходять до них від зовнішніх процесів, якими вони керують.

ПРИКЛАД УПРАВЛЯЮЧОГО АЛГОРИМА

управляючий алгоритм для керування роботом у завданні збирання предметів з конвеєра:

1. Отримання вхідних даних: положення та типи предметів, які проходять перед роботом на конвеєрі.
2. Аналіз ситуації: які предмети потрібно зібрати та в якому порядку
3. Вибір дії: де робот має рухатися і які дії виконувати, щоб зібрати предмети.

ПРИКЛАД УПРАВЛЯЮЧОГО АЛГОРИМА



4. Керування маніпулятором: відповідні команди до робота для виконання обраної дії.
5. Перевірка результатів та корекція стратегії, якщо потрібно.

СПОСОБИ ОПИСУ АЛГОРИТМІВ

- Словесний - письмовий чи усний
- Графічний (блок-схемами, кресленнями, рисунками)
- Кодований - у зашифрованому виді, зрозумілому для виконавця.
- програмний

Таблиця розрахунку заробітної плати:

| № | Прізвище | Місячна ставка | Робочих днів в місяці | Денний заробіток | Відпрацьовані дні | Заробітна платня за місяць | Сума податку (23%) | До видачі |
|-----|----------|----------------|-----------------------|------------------|-------------------|----------------------------|--------------------|-----------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| (1) | (2) | (3) | (4) | (3):(4) | (6) | (7)*(6) | (7)*0,23 | (7-8) |
| 1 | Асєєв Б. | 420 | 24 | 17,5 | 24 | 420 | 96,6 | 323,4 |
| 2 | Волін О. | 420 | 24 | 17,5 | 20 | 350 | 80,5 | 269,5 |
| 3 | Котик І. | 360 | 24 | 15,0 | 23 | 345 | 79,35 | 265,65 |

ПРОСТІ ЧИСЛА

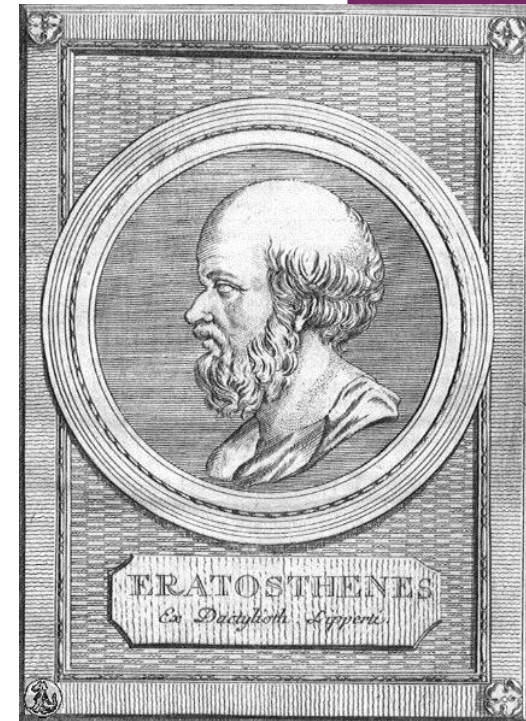
- Просте число ділиться націло лише на 1 та на себе

| 0 | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

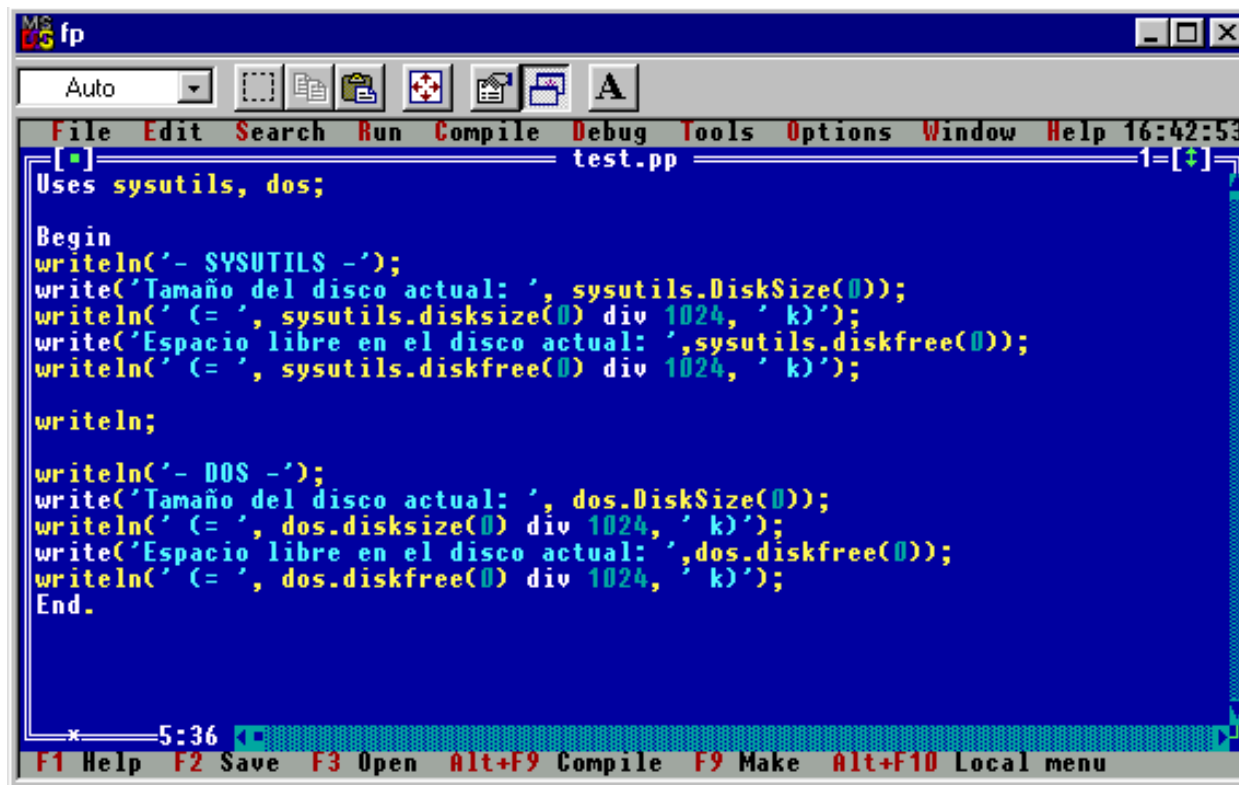
ПРИКЛАД СЛОВЕСНОГО ОПИСУ.

АЛГОРИТМ ЕРАТОСФЕНА ЗНАХОДЖЕННЯ ПРОСТИХ ЧИСЕЛ

1. Виписати всі натуральні числа від 1 до N . Викреслити 1.
2. Підкреслити найменше з невідмічених чисел.
3. Викреслити всі числа, кратні підкресленому на попередньому у кроці.
4. Якщо в списку є ще не відмічені числа, то перейти до кроку 2.
5. Всі підкреслені числа - прості.



ПРИКЛАД ПРОГРАМНОГО ОПИСУ



```
MS fp
Auto
File Edit Search Run Compile Debug Tools Options Window Help 16:42:53
test.pp
Uses sysutils, dos;

Begin
writeln('- SYSUTILS -');
write('Tamaño del disco actual: ', sysutils.DiskSize(0));
writeln(' (= ', sysutils.disksize(0) div 1024, ' k)');
write('Espacio libre en el disco actual: ', sysutils.diskfree(0));
writeln(' (= ', sysutils.diskfree(0) div 1024, ' k)');

writeln;

writeln('- DOS -');
write('Tamaño del disco actual: ', dos.DiskSize(0));
writeln(' (= ', dos.disksize(0) div 1024, ' k)');
write('Espacio libre en el disco actual: ', dos.diskfree(0));
writeln(' (= ', dos.diskfree(0) div 1024, ' k)');
End.
5:36
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

ЩО ОЗНАЧАЄ ПОБУДУВАТИ АЛГОРИТМ?

1

- Здійснити аналіз запропонованого завдання, предметної області чи інформаційної моделі, визначити проблему тощо



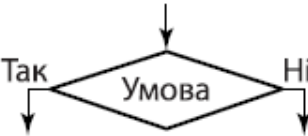
2

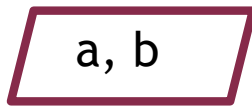
- Продумати план майбутньої діяльності щодо розв'язування завдання, вирішення проблеми чи досягнення поставленої мети

3

- Зафіксувати його за допомогою деякої системи позначень, так званою формальною мовою

ОСНОВНІ БЛОКИ АЛГОРИТМІВ

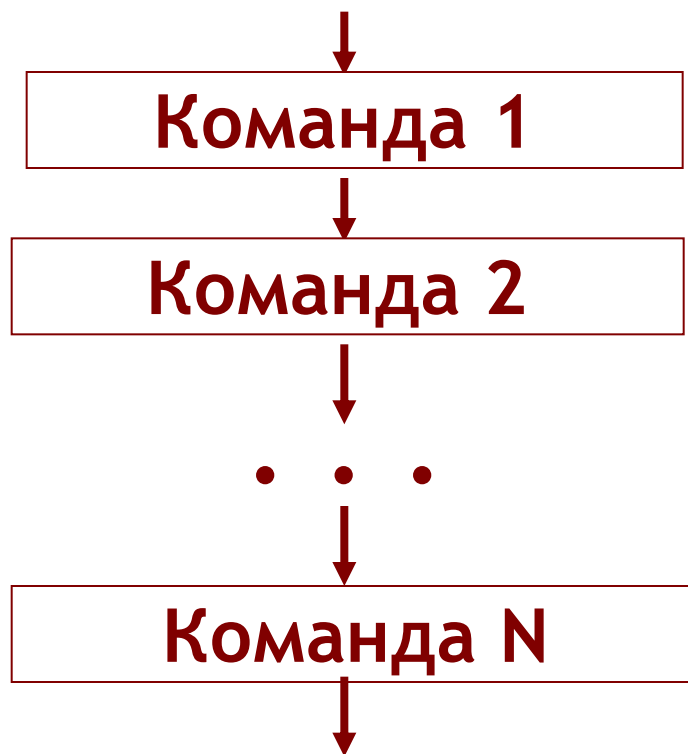
| Назва | Позначення | Призначення |
|-------------------|--|---|
| Початок алгоритму |  | <i>Точка входу</i> в алгоритм. Вказує на інструкцію, що має виконуватися першою |
| Кінець алгоритму |  | <i>Точка виходу</i> з алгоритму — блок, що завершує його виконання |
| Лінійний блок |  | Будь-яка інструкція, що не потребує перевірки умов та введення/виведення даних |
| Розгалуження |  | Інструкція, що визначає хід виконання алгоритму залежно від істинності деякої умови |



Введення, виведення даних

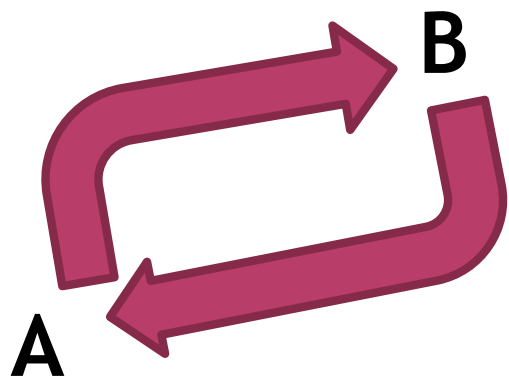
БАЗОВІ СТРУКТУРИ АЛГОРИТМІВ

1. Лінійні (прості)



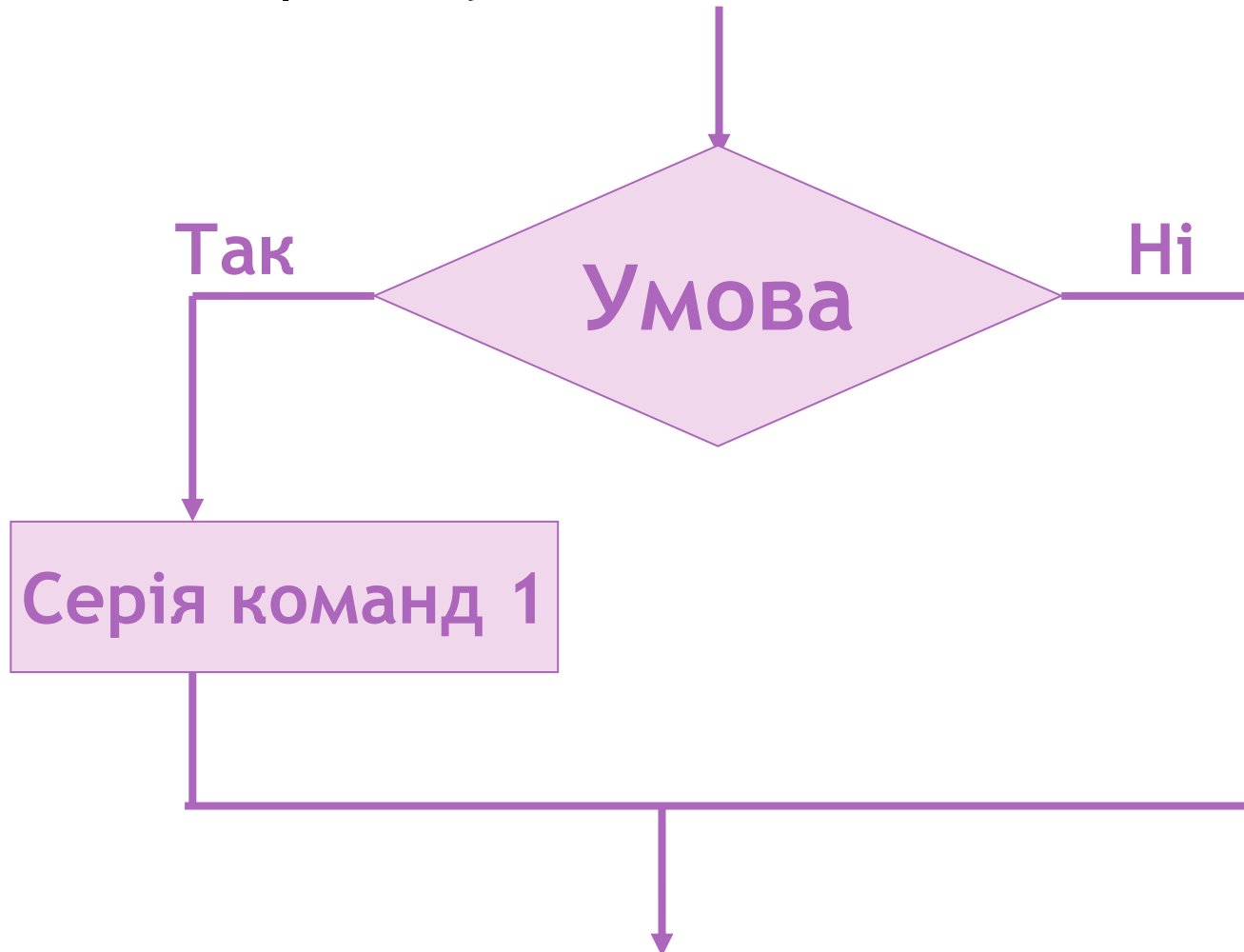
ПРИКЛАД

- Якою буде схема алгоритму, що міняє значення комірок A та B місцями, щоб тепер A зберігало старе значення B, а B - старе значення A?



БАЗОВІ СТРУКТУРИ АЛГОРИТМІВ

2 а. Неповне розгалуження

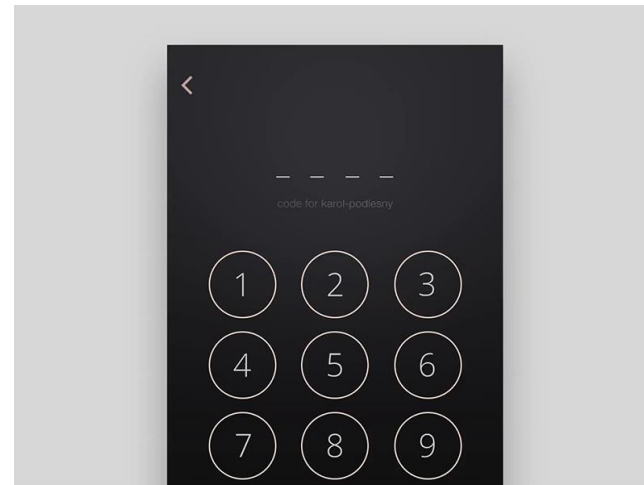


ПРИКЛАД. ПЕРЕВІРКА ПРАВОПИСУ В ТЕКСТОВОМУ РЕДАКТОРІ



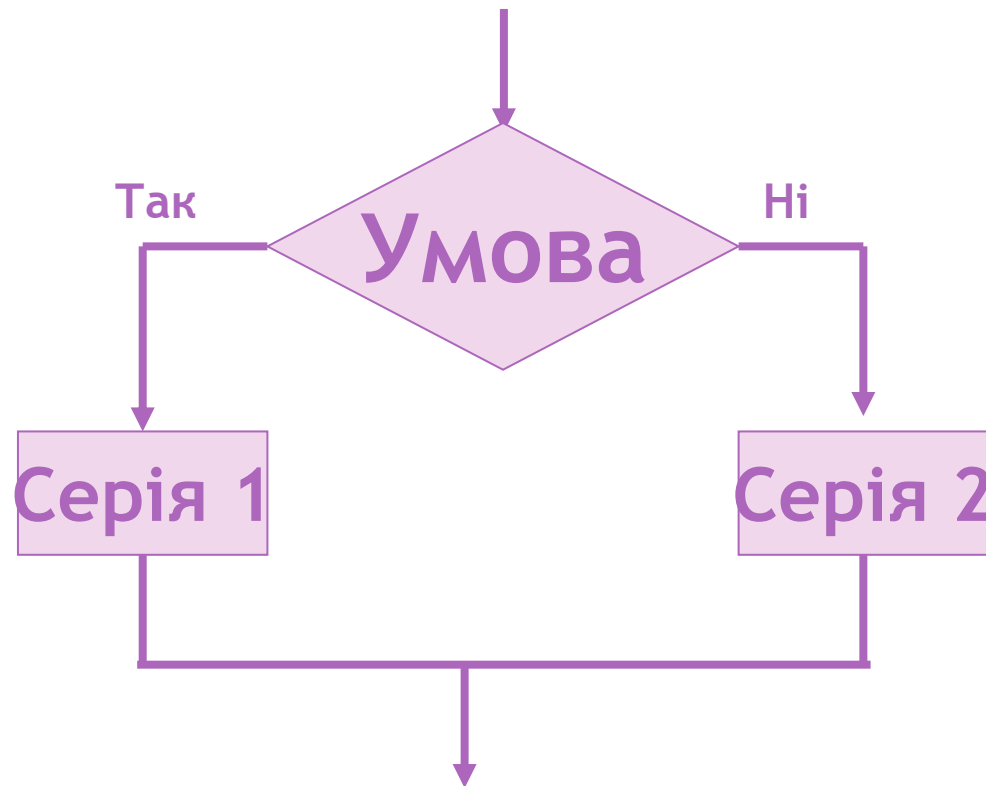
ПРИКЛАД АЛГОРИТМУ З РОЗГАЛУЖЕННЯМ

- Як виглядає схема алгоритму перевірки пін-коду до банківської картки?
- Як вона зміниться, якщо пін можна вводити до 3 разів, а потім картка блокується?



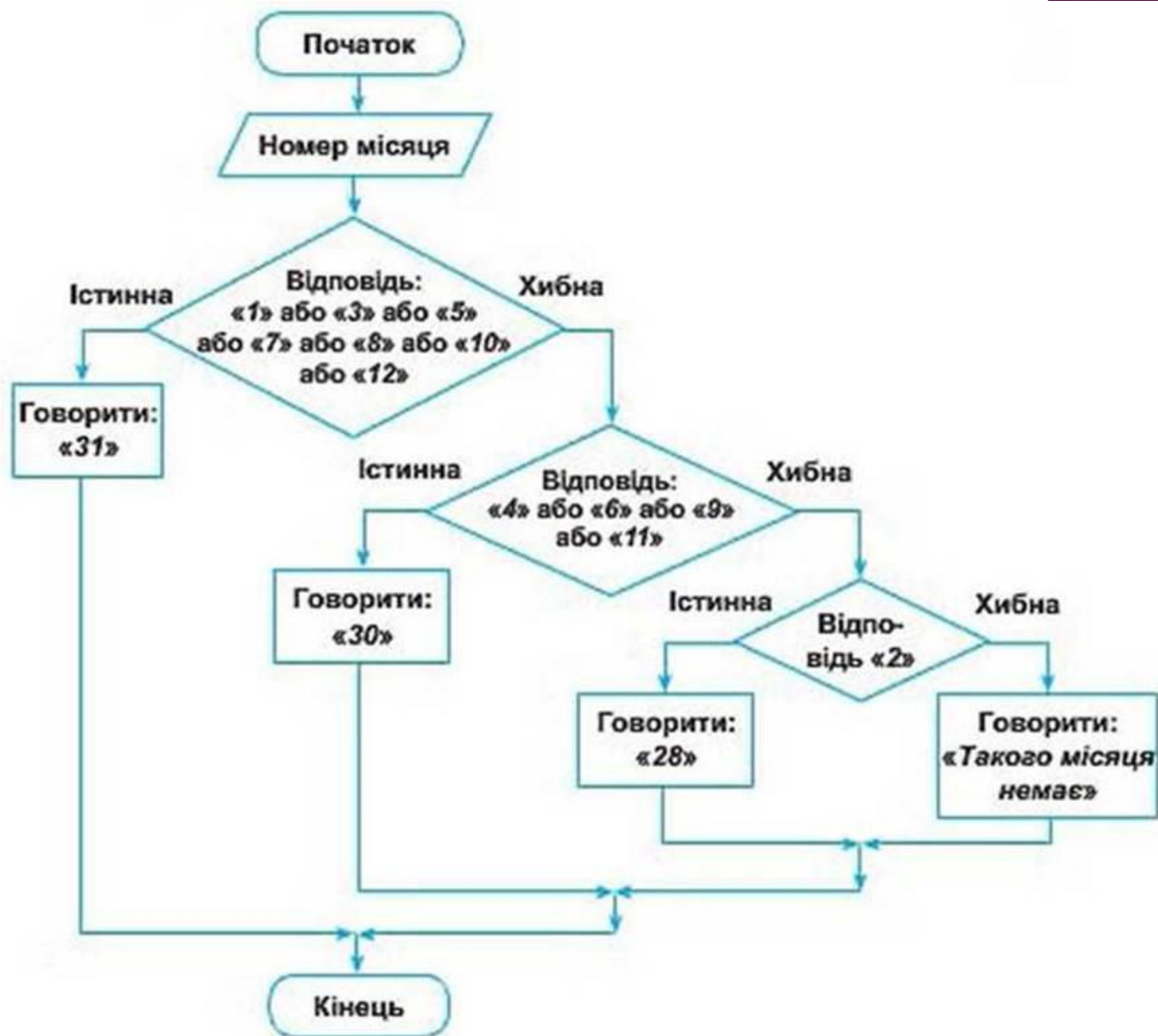
БАЗОВІ СТРУКТУРИ АЛГОРИТМІВ

2 б. Повне розгалуження



ПРИКЛАД

Алгоритм
Визначення
днів у місяці,
по номеру
місяця



Які результати
цього
алгоритму,
якщо введено 2?
Чи завжди він
вірно
працюватиме?

ЦИКЛИ

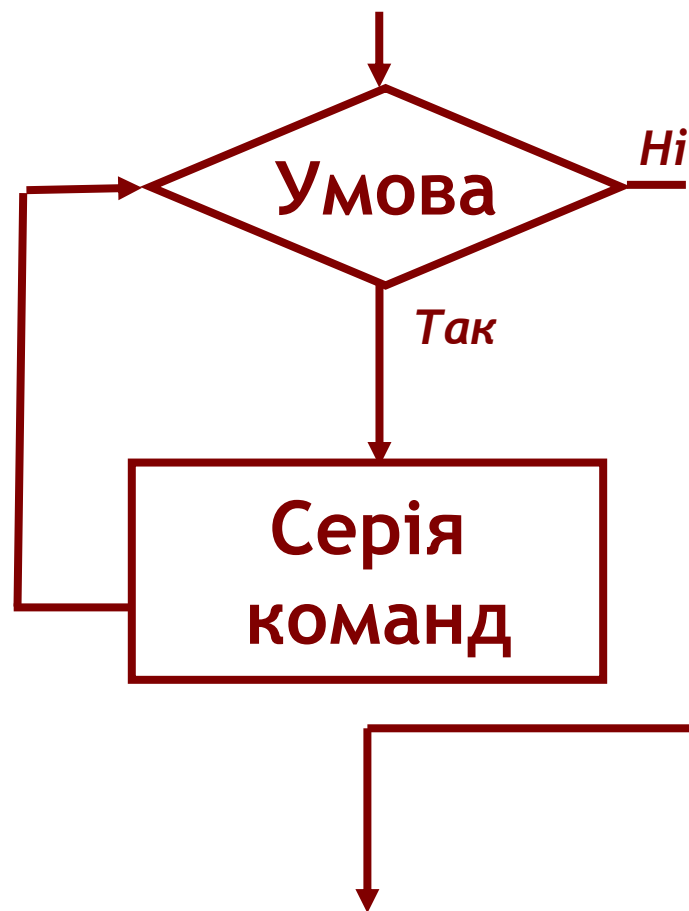
Цикл - повторення певної групи процесів (серії) за умовою.

Тіло циклу-"поки" може не виконатися жодного разу, у тому випадку, коли умова відразу ж є хибною.



БАЗОВІ СТРУКТУРИ АЛГОРИТМІВ

- Цикл з передумовою



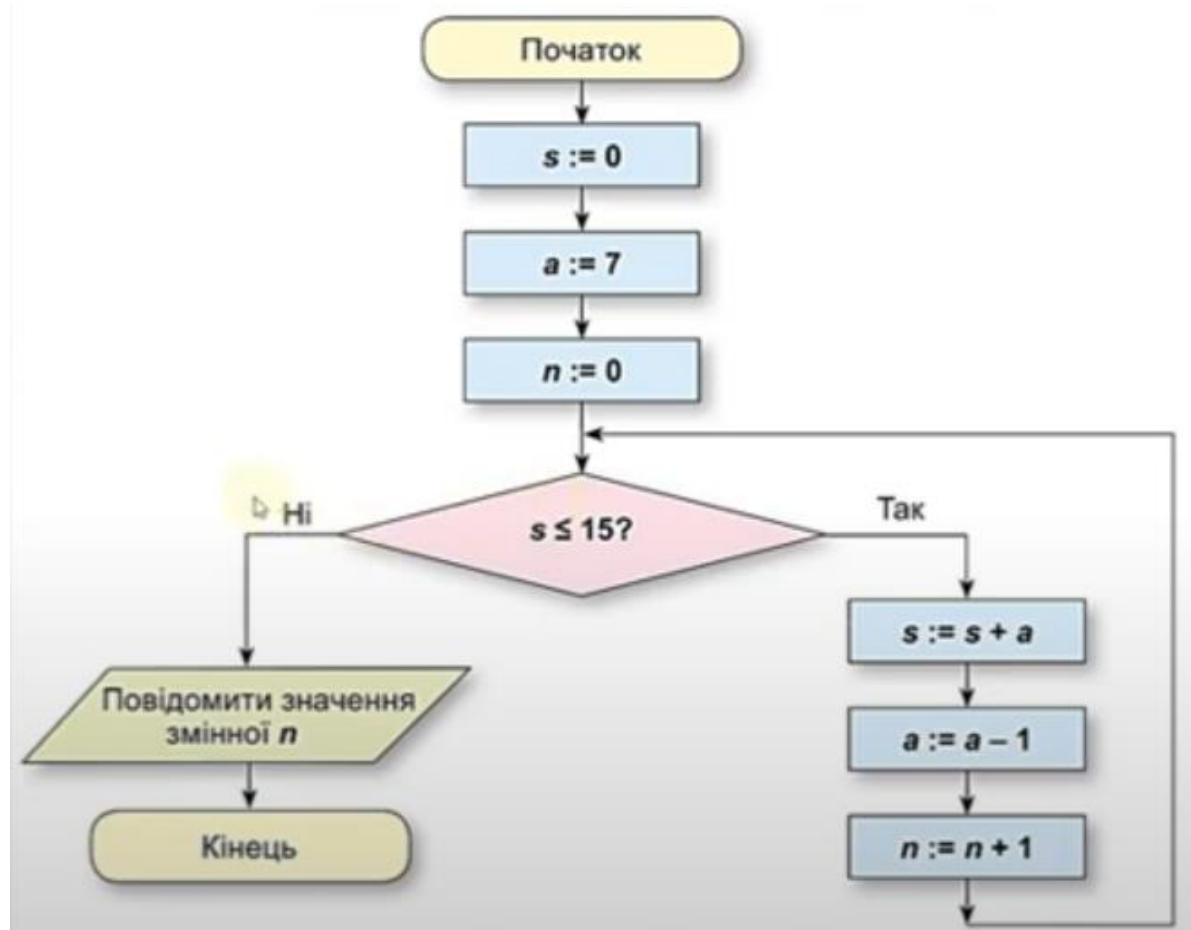
ПРИКЛАД

- Тур. група за першу годину пройшла 7 км, за кожну наступну - на 1 км менше, ніж попередню годину.
- Побудуйте алгоритм, що визначить, за скільки годин туристи пройдуть 15 км?



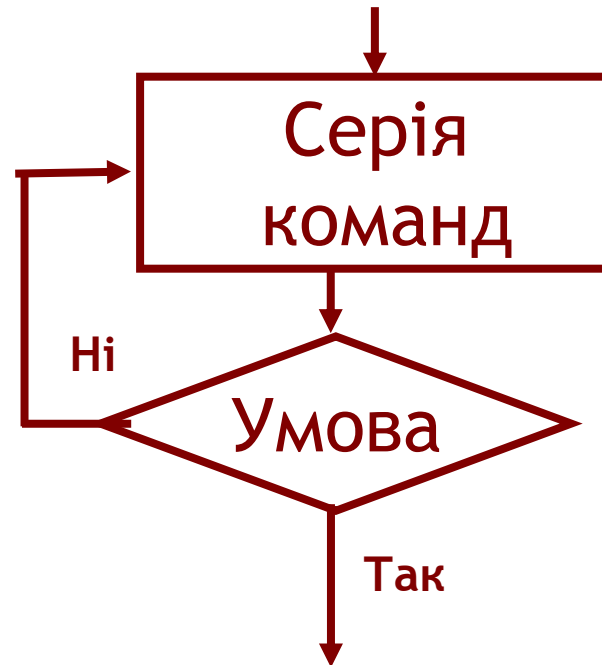
ВІДПОВІДЬ - СХЕМА АЛГОРИТМУ ПРО ТУРИСТІВ

n - кількість
годин
 a - скільки
пройшли за
поточну годину
 S - скільки
пройдено
усього на зараз



БАЗОВІ СТРУКТУРИ АЛГОРИТМІВ

- Цикл з післяумовою



ПРИКЛАД ЦИКЛА З ПІСЛЯУМОВОЮ

- ◎ **Генерація випадкового пароля:**
 - Алгоритм може використовувати цикл для створення випадкового пароля заданої довжини.
 - післяумова для перевірки стійкості пароля, наприклад, чи містить він символи нижнього та верхнього регістру, цифри та спеціальні символи.

ВИСНОВКИ

- ◉ Алгоритми та схеми алгоритмів є важливими інструментами в інформатиці та інших галузях.
- ◉ Алгоритми допомагають систематизувати та вирішувати різноманітні завдання, надають послідовність кроків, необхідних для досягнення певної мети.
- ◉ Застосування алгоритмів дозволяє автоматизувати повторювані процеси та знижує ризик помилок при виконанні завдань.
- ◉ Ефективні алгоритми дозволяють виконувати завдання швидше та ефективніше
- ◉ Розробка алгоритмів допомагає розробникам ПЗ розуміти, як потрібно реалізувати певні функції та функціонал.
- ◉ Вивчення алгоритмів допомагає людям розвивати логічне мислення, аналітичні навички та вміння розв'язувати складні завдання.

ДЯКУЮ ЗА УВАГУ!

