

## ЛЕКЦІЯ 2. СПОСОБИ ПОДАННЯ АЛГОРИТМІВ

### План

1. Словесний спосіб
  2. Формула
  3. Розрахункова таблиця
  4. Блок-схема
- 

Для подання алгоритмів застосовуються різні способи. Кожний з них надає певні засоби для опису дій, які треба виконати, та встановлення послідовності їх виконання.

1. У повсякденному житті найчастіше застосовується *словесний спосіб*. Алгоритм подається як послідовність окремих занумерованих пунктів, кожний з яких містить команду на виконання певної дії. Команди записуються словами. Пункти виконуються один за одним у порядку зростання їх номерів, якщо немає спеціальної вказівки на перехід до виконання іншого пункту, номер якого задається. Словесний спосіб подання алгоритму є найбільш прийнятним для опису інструкцій побутового характеру, дій на випадок надзвичайної ситуації, фармакологічних чи кулінарних рецептів, тощо.

### Приклад 1

1) Завдання : закип'ятити чайник

#### Алгоритм:

1. Налити в чайник води.
2. Запалити газову конфорку.
3. Поставити чайник на плиту.

2) Завдання: обчислити вираз  $2*(2+2)$

#### Алгоритм:

1. Додати  $2+2$  і запам'ятати результат 4.
2. Виконати множення  $2*4$  і запам'ятати результат 8 .

*Зауваження:* якщо змінити послідовність команд, алгоритм може бути не правильним.



## Алгоритм приготування бутерброду



1. Взяти шматок хліба.
2. Покласти на хліб листок салату.
3. Покласти *зверху* котлету.
4. Покласти *зверху* шматок сиру.
5. Покласти на сир шматок помідора.
6. Накрити *іншим шматком хліба*



### Рецепт "Піца нашвидкуруч":

1. В рівних пропорціях змішайте сметану з томатною пастою, за бажанням додайте орегано чи інші спеції.
2. Отриманий соус намажте на хліб.
3. Зверху викладіть порізані на шматочки помідори та болгарський перець, половинки маслин. Або інші улюблені варіанти начинки.
4. Натертий чи нарізаний скибочками сир покласти зверху.
5. На кілька хвилин поставити отримане в духовку, аерогриль чи мікрохвильовку.



### Діамо

**Запишіть** словесні алгоритми:

1. Як приготувати ямайський джерк.
2. Як приготувати ягідний смузі.
3. Як приготувати салат «Цезар».
4. Як приготувати гаспаччо.
5. Як приготувати терамісу.
6. Як приготувати хачапурі.
7. Як приготувати англійський пудинг.
8. Як приготувати ролл Червоний дракон
9. Як приготувати равіолі.

Завдання виконати у файлі *Word* (назва файлу – **Приклади словесних алгоритмів**).

2. Записом алгоритму можна вважати **формулу**, тому що з неї впливає порядок здійснення обчислень для здобуття числового результату.

Наприклад, обчислення площі трикутника за трьома сторонами:

1. Обчислити півпериметр за формулою:

$$p = \frac{a+b+c}{2} - \text{півпериметр}$$

2. Обчислити площу за формулою:

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$



**Запишіть** алгоритми обчислення площ наступних фігур у вигляді формули:

1. Формули площі трикутника
2. Формули площі квадрата
3. Формула площі прямокутника
4. Формули площі паралелограма
5. Формули площі ромба
6. Формула площі трапеції
7. Формули площі опуклого чотирикутника
8. Формули площі круга
9. Формула площі еліпса

**Завдання виконати у файлі Word (назва файлу – Приклади алгоритмів).**

3. Якщо виконується серія розрахунків за однаковими формулами, то для запису алгоритму іноді використовується **розрахункова таблиця**, де визначаються всі етапи обчислень і фіксуються проміжні результати.

**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	Асеев 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

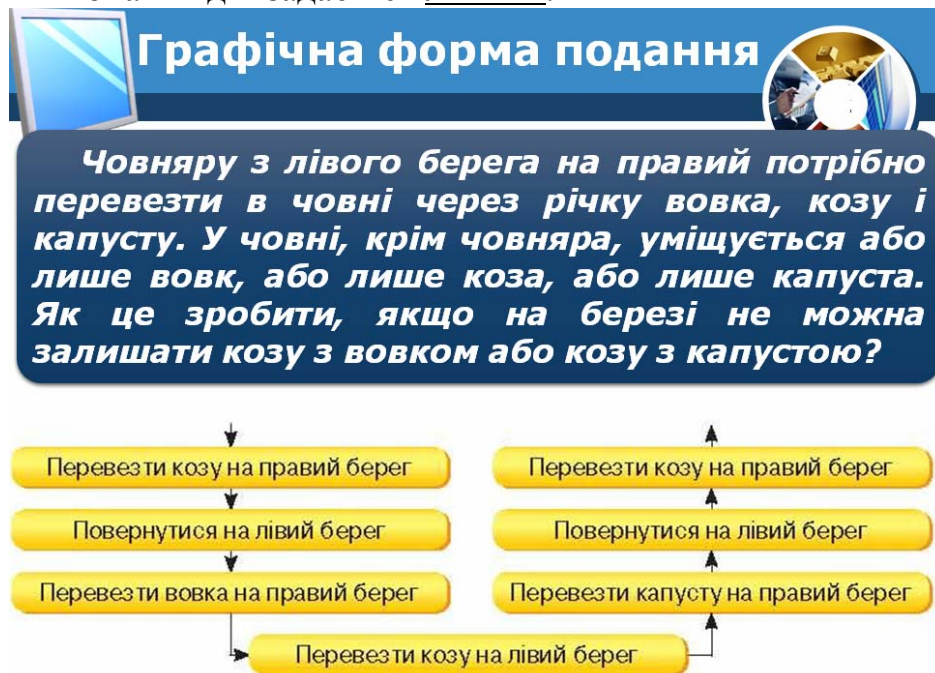


Записати алгоритм підрахунку калорій для різних вікових категорій у вигляді розрахункової таблиці. Додаткову інформацію знайти в Інтернеті.

Завдання виконати у файлі *Word* (назва файлу – **Приклад алгоритму-таблиці**)

4. Поширеним способом наочного подання алгоритму є **блок-схема (графічна форма подання)**. Вона складається з геометричних фігур, які з'єднані напрямленими лініями. Зміст дій описується всередині геометричних фігур.

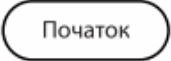

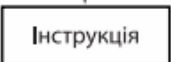

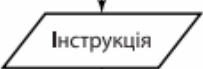
Порядок виконання дій задається лініями.




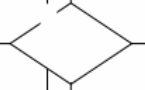
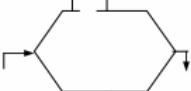
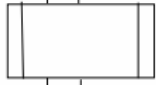




Для подання алгоритму застосовуються геометричні фігури двох видів - прямокутники та ромби. У прямокутниках записують дії, які мають виконуватися, в ромбах - умови, які треба перевіряти. перевірка умов потрібна для вибору тих чи інших подальших дій. Якщо прямокутник має один вхід і один вихід, то у ромба вхід один, а виходів - два. Коли перевіряється умова, записана в ромбі, то існує два можливих варіанти: умова або виконується, або ні. На ці випадки ("так" і "ні") і є два виходи з ромба, які спрямовують подальші дії виконавця алгоритму залежно від результату перевірки умови.

Для того, щоб алгоритми, подані блок-схемами, було зручно читати та виконувати, введено допоміжні елементи: овали, які застосовуються для позначення початку або кінця алгоритму, та паралелограми - для введення або виведення даних.

## ОСНОВНІ БЛОКИ АЛГОРИТМУ

Назва	Позначення	Призначення
Початок алгоритму		Точка входу в алгоритм. Вказує на інструкцію, що має виконуватися першою
Кінець алгоритму		Точка виходу 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		Інструкція з уведення даних в інформаційну систему чи виведення даних з неї

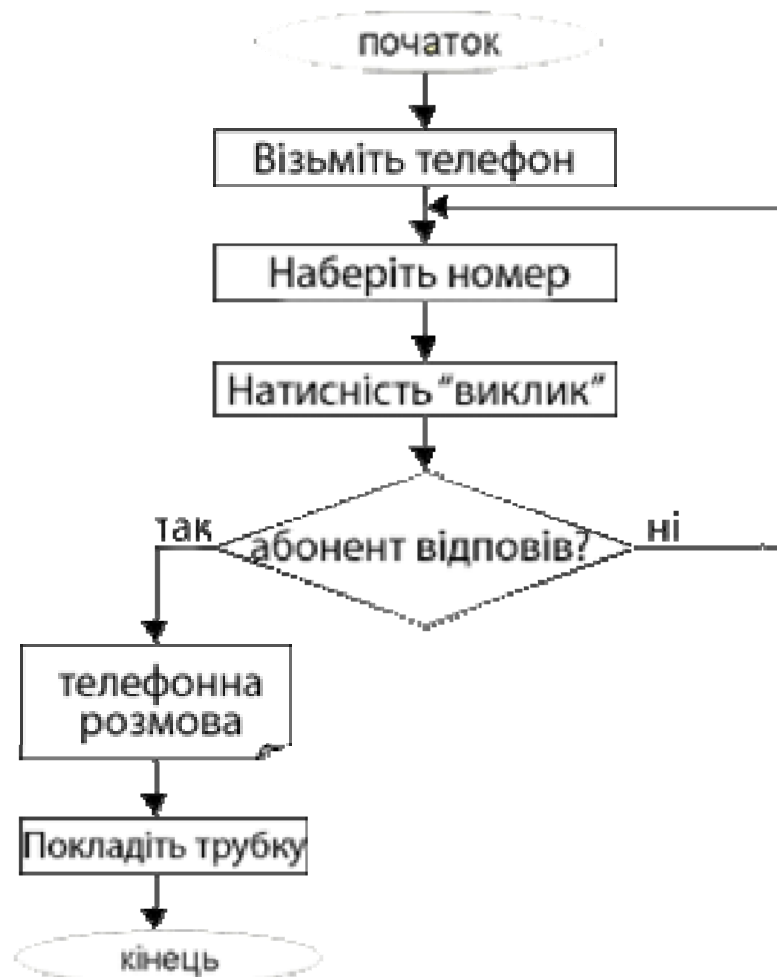
Для складних алгоритмів використовується більш розширена таблиця:

	Обчислювальна дія чи послідовність обчислювальних дій
	Перевірка умов
	Початок циклу
	Обчислення за підпрограмою чи стандартною підпрограмою
	Ввід чи вивід даних
	Розрив лінії потоку, з'єднувач
	Початок, кінець, останов, вхід і вихід у підпрограмах
	Пояснення, зміст підпрограм, формули

З алгоритмами ми зустрічаємось на кожному кроці. Будь-яка людина користується набором всіляких алгоритмів - правил, які вона засвоює в процесі виховання та навчання, фізичного тренування, набуття власного життєвого досвіду. Інструкції користування ліфтом, телефоном, різноманітними автоматами і побутовими приладами, правила переходу вулиці, надання першої медичної допомоги, приготування страв, виконання арифметичних дій або гімнастичних вправ, покроковий план розв'язування задачі або проведення фізичного чи хімічного досліду – все це можна вважати алгоритмами. Алгоритми відіграють велику роль у житті людини: вони зберігають її сили та час, тому що саме через алгоритми передається здобутий досвід ефективних способів діяльності. Один раз засвоєним правилом людина може користуватися все своє життя.

### Приклад.

#### Алгоритм телефонування з мобільного телефону:



#### ОСНОВНІ ТИПИ БЛОК-СХЕМ

- 1) Лінійна
- 2) Розгалуження
- 3) Цикл (поки)
- 4) Цикл (для)

Розглянемо та порівняємо алгоритми відкриття та збереження текстового документа у програмі **Блокнот**. Текстові описи алгоритмів наведено на рисунку:

1. Вибрати меню Файл.
2. Виконати команду Відкрити.
3. У вікні Відкрити вибрати файл, що відкриватиметься.
4. Клацнути кнопку Відкрити.

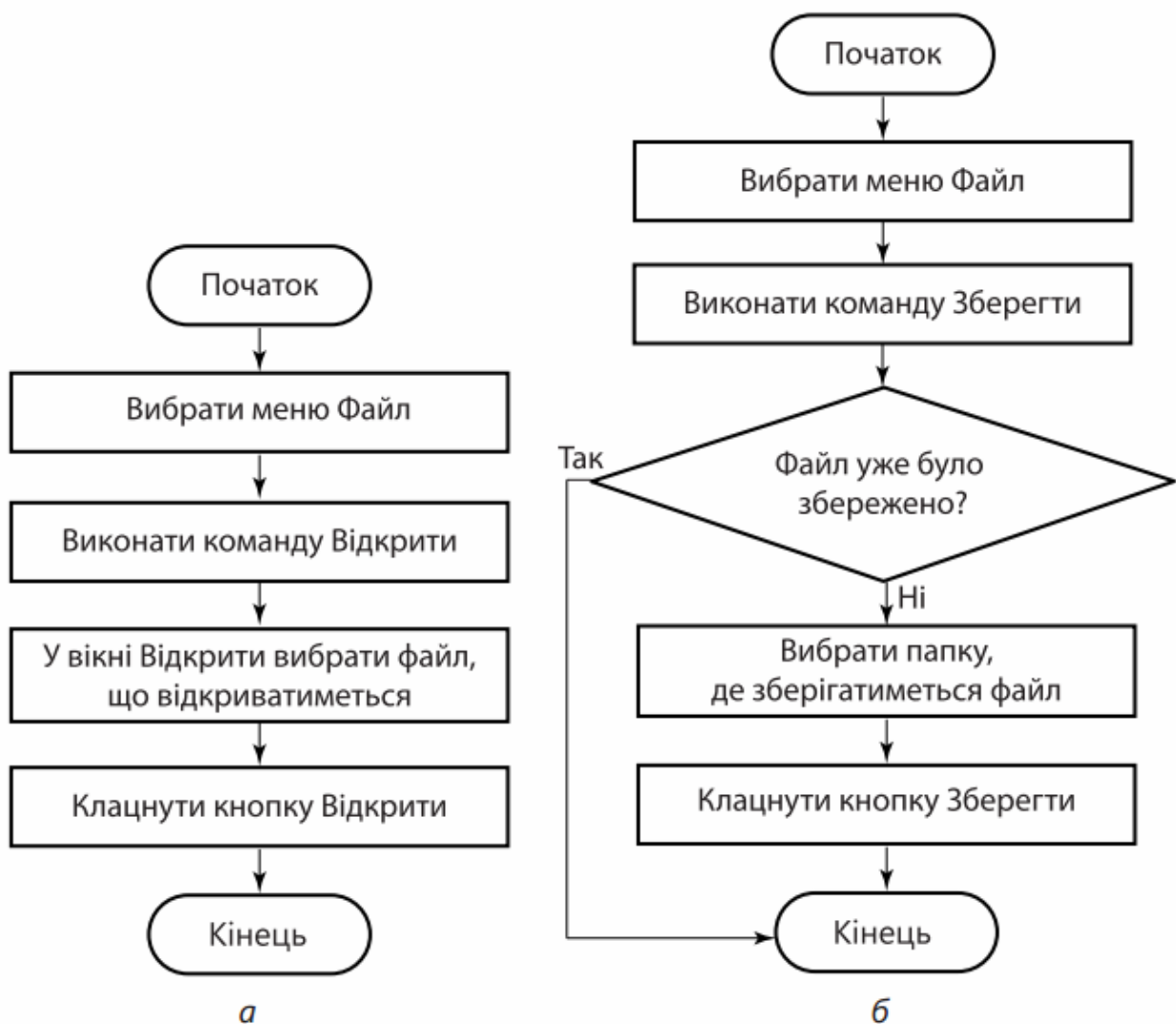
*a*

1. Вибрати меню Файл.
2. Виконати команду Зберегти.
3. Якщо файл ще не було збережено, то виконати кроки 4–6.
4. Вибрати папку, де зберігатиметься файл.
5. Увести ім'я файлу.
6. Клацнути кнопку Зберегти.

*б*

Текстові описи алгоритмів роботи з програмою Блокнот:  
*a* — відкриття документа; *б* — збереження документа

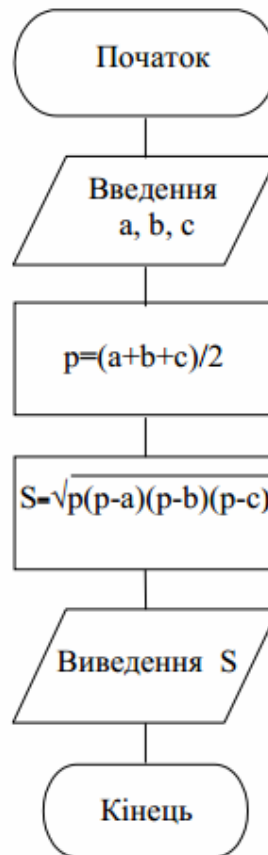
Структурна відмінність між цими алгоритмами очевидна: в алгоритмі відкриття документа завжди виконуються всі кроки, а чи будуть виконані кроки 4–6 в алгоритмі збереження, залежить від того, чи вперше зберігають документ. Алгоритм, усі інструкції якого виконуються безумовно, називають *лінійним*. Алгоритм, у якому виконання чи невиконання певних інструкцій залежить від того, чи справджується деяка умова, отримав назву *алгоритм із розгалуженням*.



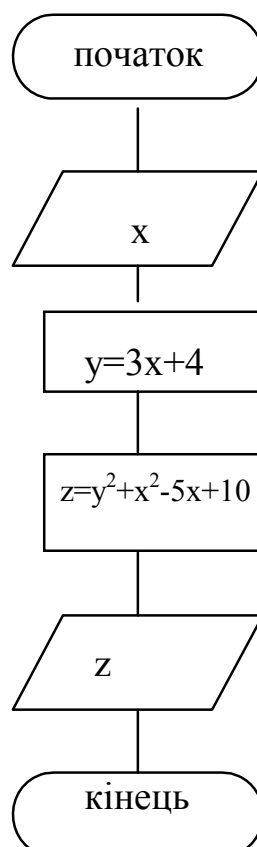
Графічне подання алгоритмів роботи з програмою Блокнот:  
*a* — відкриття документа; *б* — збереження документа

Отже, *лінійним називається алгоритм* (фрагмент алгоритму), в якому окремі команди виконуються послідовно одна за одною, не залежно від значень вхідних даних і проміжних результатів.

**Приклад** алгоритму лінійної структури (обчислення площі трикутника за формулою Герона):



**Приклад.** Знайти значення виразу:  $Z = y^2 + x^2 - 5y + 10$ , де  $y = 3x + 4$

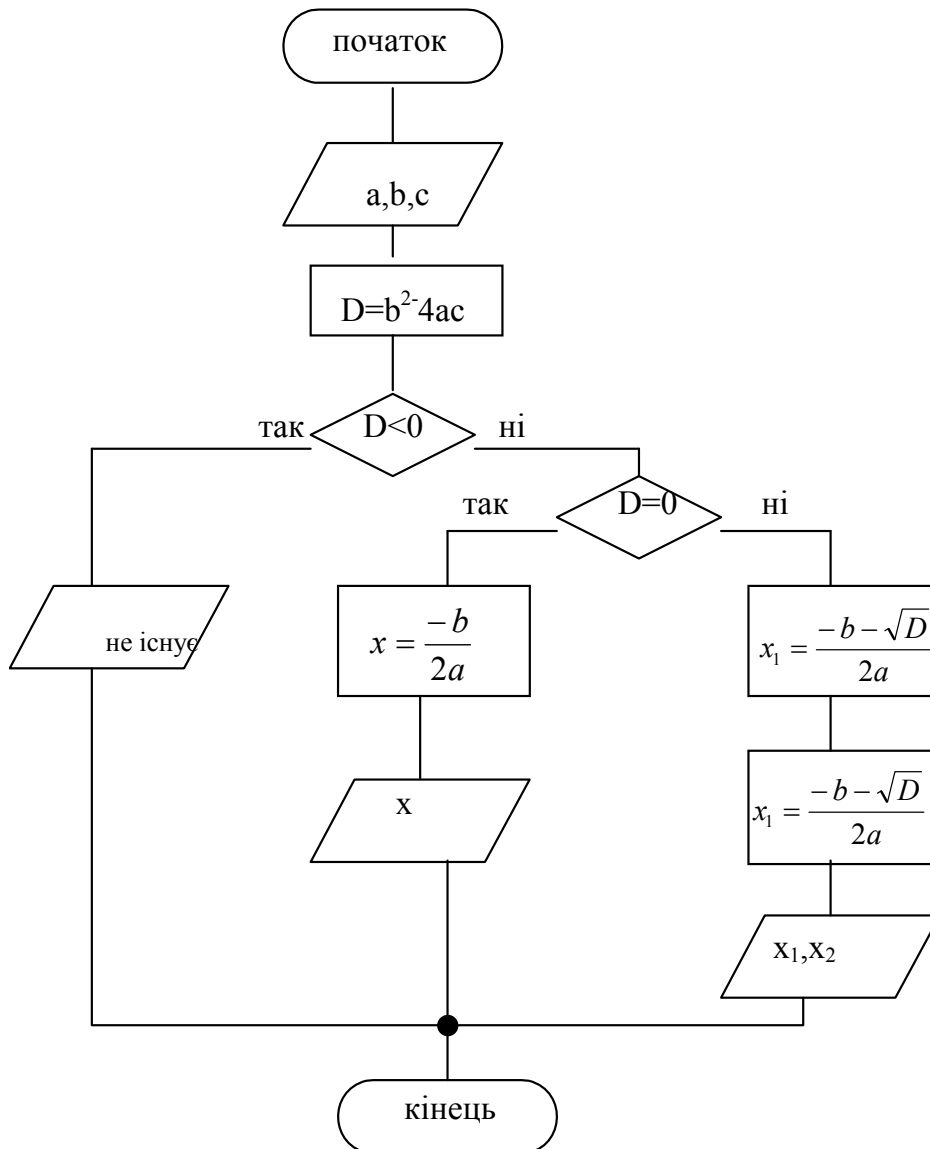




Алгоритм, у якому виконання чи невиконання певних інструкцій залежить від того, чи справджується деяка умова, отримав назву *алгоритм із розгалуженням*.

**Приклад розгалуженого алгоритму.**

Розв'язати квадратне рівняння  $ax^2+bx+c=0$



## Цикл (поки)

### Приклад.

Знайти  $n!$

$$1! = 1$$

$$2! = 1 * 2 = 2$$

$$3! = 1 * 2 * 3 = 6$$

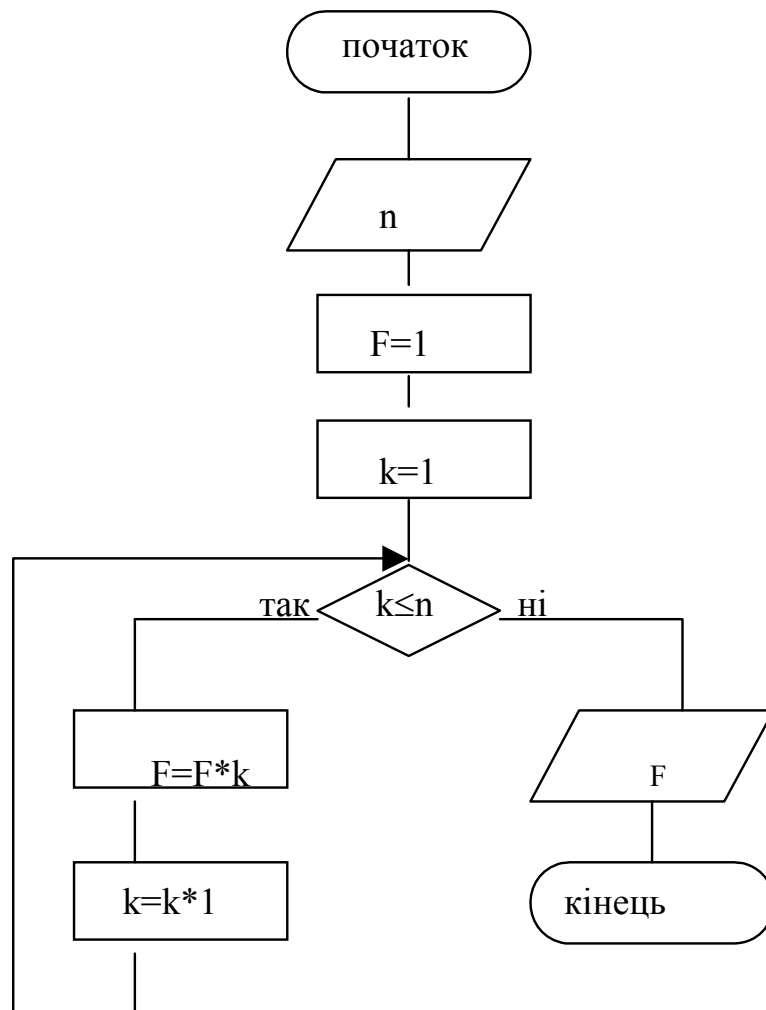
$$4! = 1 * 2 * 3 * 4 = 24$$

.

.

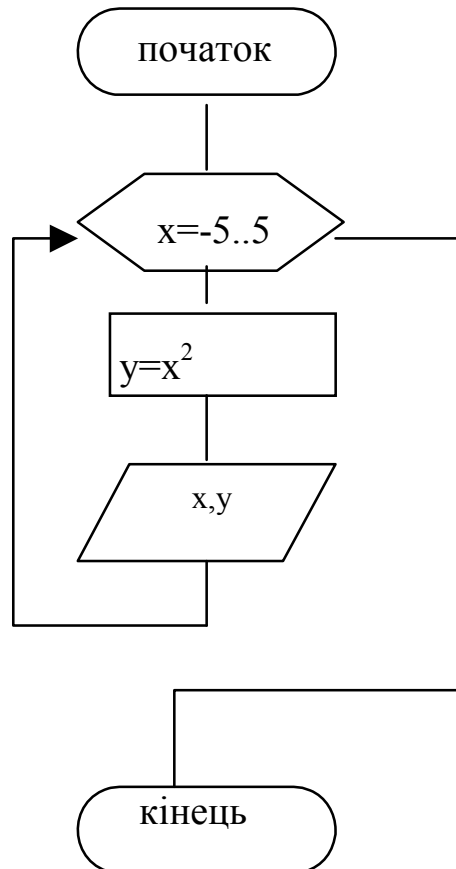
.

$$n! = 1 * 2 * 3 * 4 * \dots * (n-1) * n$$



### Цикл (для).

Вивести на екран значення функції  $y=x^2$  для  $x$  від  $-5$  до  $5$  з кроком  $1$ .



Скласти блок-схему алгоритму:

- знаходження суми  $S$  трьох чисел  $a, b, c$ ;
- перехід дороги;
- обчислення функції

$$y = \begin{cases} \sin x, & x < 0 \\ \cos x, & x \geq 0 \end{cases}$$

- обчислення суми

$$S = \sum_{i=1}^n a_i$$

- обчислення дійсних коренів квадратичного рівняння;
- знаходження значення виразу:  $Z=y^2+x^2-5y+10$ , де  $y=3x+4$ ;
- зміни пробитого колеса у автомобіля;
- створити блок-схему для обчислення функції  $X*\text{SIN}(X/2)$  по введеному значенню аргумента  $X$