

Прості завдання

Виведення даних за допомогою вказівки `print()`

- використовується лише `print()`,
- без f-рядків (`f"""`) та `.format()`,
- без створення змінних.

1. Виведи на екран слово "Привіт".
2. Виведи на екран фразу "Python — це круто!".
3. Виведи число 42.
4. Виведи вираз `"2 + 3 = 5"`, використовуючи лише літерали.
5. Виведи на екран два слова в одному рядку через пропуск: "Good" та "morning".
6. Виведи трикрапку: "...".
7. Виведи символи "@#\$\$%" в одному рядку.
8. Виведи порожній рядок (тобто просто перейди на новий рядок).
9. Виведи на екран вираз `"100 / 2 = 50"`.
10. Виведи на екран назву мови програмування у лапках: "Python".
11. Виведи на екран "Yes" тричі поспіль у одному рядку через кому.
12. Виведи на екран фразу "Я люблю програмувати!".
13. Виведи суму `7 + 8` у вигляді тексту: `"7 + 8 = 15"`.
14. Виведи на екран трикутник з зірочок, використовуючи кілька викликів `print()`:
15. Виведи на екран "Start", а потім на новому рядку — "Finish".
16. Виведи на екран символ табуляції між двома словами: "Left", "Right".
17. Виведи на екран вираз `"3 * 3 = 9"`.
18. Виведи на екран текст "Мій друг - комп'ютер!", правильно використавши апостроф.
19. Виведи на екран літери "A B C" через один пропуск.
20. Виведи на екран вираз `"5 - 2 = 3"`.

Кожне завдання можна виконати одним або кількома викликами `print()` з літералами (рядковими або числовими), не використовуючи змінних, f-рядків чи `.format()`.

Використовуємо змінні

Завдання для вивчення Python, що відповідають умовам:

використання змінних та `print()`,
без f-рядків (`f"""`) і методу `.format()`,
дозволяється використовувати конкатенацію рядків (+) або передавати кілька аргументів у `print()`.

1. Збережи число 10 у змінній `x` і виведи його.
2. Збережи рядок "Hello" у змінній `greeting` і виведи його.

3. Збережи два числа: `a = 7`, `b = 3`. Виведи їх суму (використовуй `print(a, "+", b, "=", a + b)`).
4. Збережи ім'я у змінній `name` і виведи "Мене звати " + значення змінної.
5. Збережи три числа в окремих змінних (`x`, `y`, `z`) і виведи їх через кому.
6. Створи змінну `city = "Київ"` і виведи фразу "Столиця України - Київ", використовуючи змінну.
7. Збережи булеве значення `True` у змінній `is_student` і виведи його.
8. Створи дві змінні: `first = "Python"`, `second = "rocks!"`. Виведи їх разом у одному рядку.
9. Збережи число 3.14 у змінній `pi` і виведи "Значення π приблизно дорівнює 3.14", використовуючи змінну.
10. Створи змінну `score = 95` і виведи "Ваш рахунок: 95".
11. Збережи два слова у змінних: `word1 = "Сонячний"`, `word2 = "день"`. Виведи "Сонячний день".
12. Присвой змінній `temperature = -5` і виведи "Температура: -5 °C".
13. Збережи вік у змінній `age = 14` і виведи "Мені 14 років".
14. Створи дві числові змінні `width = 10`, `height = 20`. Виведи повідомлення у вигляді "Прямокутник: 10 x 20".
15. Збережи три літери у змінних `a = "A"`, `b = "B"`, `c = "C"` і виведи їх у вигляді "A-B-C".
16. Присвой змінній `price = 199` і виведи "Ціна: 199 грн.".
17. Створи змінну `language = "Python"` і виведи "Я вивчаю Python".
18. Збережи два числа `x = 100`, `y = 50` і виведи їх різницю у форматі "100 - 50 = 50".
19. Збережи у змінній `planet = "Україна"` і виведи "Моя країна - Україна".
20. Створи змінну `completed = False` і виведи "Завдання виконано: False".

Ці завдання допомагають зрозуміти, як працюють змінні, як передавати кілька аргументів у `print()` (з автоматичним розділенням пропусками) і як використовувати конкатенацію, не застосовуючи сучасні способи форматування рядків.

Літерали та змінні, з використанням f-рядків

Завдання для вивчення Python:

використання літералів, змінних та f-рядків (f""),
інші способи форматування (`str.format()`), конкатенація тощо) не
використовуються.

1. Збережи число 14 у змінній `age` і виведи: "Мені 14 років" за допомогою f-рядка.
2. Збережи ім'я "Олена" у змінній `name` і виведи: "Привіт, Олена!".

3. Збережи два числа: $a = 12$, $b = 8$. Виведи їх суму у форматі: "12 + 8 = 20".
4. Збережи ціну `price = 299.99` і виведи: "Найкраща ціна: 299.99 грн."
5. Збережи назву міста "Львів" у змінній `city` і виведи: "Львів вітає вас!"
6. Збережи три числа у змінних `x`, `y`, `z` (наприклад, 3, 5, 7) і виведи: "Numbers: 3, 5, 7".
7. Збережи булеве значення `is_raining = True` і виведи: "Чи дощитиме? True".
8. Збережи курс валют `usd_to_uah = 39.5` і виведи: "1 USD = 39.5 UAH".
9. Збережи два слова: `noun = "кіт"`, `verb = "спить"`. Виведи: "Мій кіт спить".
10. Збережи дробове число `pi = 3.14159` і виведи: " $\pi \approx 3.14159$ ".
11. Збережи кількість очок `score = 98` і виведи: "Ваш рахунок: 98".
12. Збережи два імені: `friend1 = "Ігор"`, `friend2 = "Марина"`. Виведи: "Мої друзі: Ігор та Марина".
13. Збережи температуру `temp = -10` і виведи: "Температура на вулиці: -10°C".
14. Збережи назву мови програмування `lang = "Python"` і виведи: "Мені подобається Python!".
15. Збережи кількість днів `days = 7` і виведи: "У тижні 7 днів".
16. Збережи два числа `length = 15`, `width = 10` і виведи площу прямокутника: "Площа: 150".
17. Збережи швидкість `speed = 60` і виведи: "Поточна швидкість: 60 км/год".
18. Збережи логічне значення `has_access = False` і виведи: "Доступ дозволено: False".
19. Збережи кількість студентів `students = 25` і виведи: "У нашому класі 25 учнів".
20. Збережи назву книги `book = "Сталь та сонячний шторм"` і автора `author = "Редгрейн Лебовскі"` і виведи: "Книга: Сталь та сонячний шторм, автор: Редгрейн Лебовскі".

Використання вказівки `input()` (тільки рядкові змінні)

Завдання, у яких використовується `input()` і працюють тільки з рядковими змінними (без перетворення на числа чи інші типи, без `int()`, `float()` тощо). Усі значення залишаються рядками, а виведення відбувається через `print()` — можна використовувати f-рядки, якщо це не заборонено.

1. Запитай користувача ввести своє ім'я і виведи: "Привіт, [ім'я]!"
2. Запитай користувача ввести улюблене місто і виведи: "Твоє улюблене місто — [місто]."
3. Запитай користувача ввести назву улюбленої кав'ярні і виведи: "Я теж люблю [кав'ярня]!"

4. Запитай користувача ввести ім'я друга і виведи: "[Ім'я], ти найкращий(а)!".
5. Запитай користувача ввести улюблену книгу і виведи: "Цікаво! Я читав(ла) книгу '[книга]'".
6. Запитай користувача ввести колір і виведи: "Твій улюблений колір — [колір]. Красивий вибір!".
7. Запитай користувача ввести назву улюбленої пісні і виведи: "А мені подобається пісня '[пісня]'!".
8. Запитай користувача ввести прізвище і виведи: "Вітаю, [прізвище]!".
9. Запитай користувача ввести назву улюбленого фільму і виведи: "Фільм '[фільм]' — класика!".
10. Запитай користувача ввести ім'я тварини і виведи: "Як звати твою тварину? [ім'я тварини] — чудове ім'я!".
11. Запитай користувача ввести улюблену їжу і виведи: "Смачно! Я теж люблю [їжа]".
12. Запитай користувача ввести назву школи/університету і виведи: "Ти навчаєшся в [навчальний заклад]?".
13. Запитай користувача ввести назву своєї улюбленої гри і виведи: "Граючи в '[гра]', ти точно відпочиваєш!".
14. Запитай користувача ввести слово-настрій (наприклад, "радість", "спокій") і виведи: "Сьогодні твій настрій — [настрій]".
15. Запитай користувача ввести ім'я уявного персонажа і виведи: "Персонаж на ім'я [ім'я] звучить епічно!".
16. Запитай користувача ввести назву мови програмування (як рядок!) і виведи: "Мова [мова] — чудовий вибір для початку!".
17. Запитай користувача ввести улюблену пору року і виведи: "Моя улюблена пора року — [пора року], як і в тебе!".
18. Запитай користувача ввести назву свого хобі і виведи: "Займатися [хобі] — це круто!".
19. Запитай користувача ввести назву улюбленого напою і виведи: "Хочеться випити [напій] зараз!".
20. Запитай користувача ввести одне слово-характеристику себе (наприклад, "допитливий", "креативний") і виведи: "Ти — [характеристика]! Це цінна риса."

Усі завдання передбачають, що введені дані залишаються **рядками**, навіть якщо користувач вводить цифри (наприклад, "42" обробляється як текст, а не число). Це добре підходить для початкового ознайомлення з input() та роботою з текстовими даними.

Використання вказівки input() (з int та float)

Завдання, у яких використовується input() з перетворенням на int або float, тобто користувач вводить числові дані, які програма обробляє як цілі або дійсні числа. Усі завдання передбачають використання арифметичних операцій або числового виведення.

1. Запитай вік користувача (int) і виведи: "Тобі [вік] років."
2. Запитай кількість яблук (int) і виведи: "У тебе [яблука] яблук."
3. Запитай ціну товару (float) і виведи: "Ціна: [ціна] грн."
4. Запитай довжину і ширину прямокутника (int або float) і виведи його площу: "Площа: [площа]".
5. Запитай оцінку з тесту (int, від 0 до 100) і виведи: "Твій бал: [бал]."
6. Запитай відстань у кілометрах (float) і виведи її в метрах: "[км] км = [метри] м."
7. Запитай кількість годин (int) і виведи, скільки це хвилин: "[години] годин — це [хвилини] хвилин."
8. Запитай вагу в кілограмах (float) і виведи в грамах: "[кг] кг = [грами] г."
9. Запитай ціле число (int) і виведи його квадрат: "Квадрат числа [n] — [n*n]."
10. Запитай дробове число (float) і виведи його округлене значення: "Округлено: [round(число)]."
11. Запитай два цілих числа (a, b) і виведи їх суму: "[a] + [b] = [сума]".
12. Запитай число (float) і виведи його подвоєне значення: "Подвоєне: [число * 2]".
13. Запитай висоту трикутника і довжину основи (float) і виведи площу ($0.5 * \text{основа} * \text{висота}$).
14. Запитай кількість днів (int) і виведи, скільки це тижнів і днів: "[днів] днів — це [тижнів] тижнів і [rest] днів."
15. Запитай швидкість (float, км/год) і час (float, год) і виведи пройденої відстань: "Відстань: [швидкість * час] км."
16. Запитай температуру в градусах Цельсія (float) і виведи її в градусах Кельвіна ($C + 273.15$): "[C] °C = [K] K."
17. Запитай ціле число (int) і виведи, чи воно парне: "Число [n] — парне: [True/False]."
18. Запитай кількість сторінок у книзі (int) і кількість днів на читання (int), виведи середню кількість сторінок на день: "Потрібно читати по [середнє] сторінок на день."
19. Запитай ціну (float) і відсоток знижки (int, наприклад 10), і виведи нову ціну: "Ціна зі знижкою: [нова_ціна] грн."
20. Запитай три цілих числа (a, b, c) і виведи їх середнє арифметичне: "Середнє: [середнє]".

Ці завдання допомагають опанувати:

перетворення введених даних у int або float,
базові арифметичні операції,
роботу з даними користувача як з числами, а не лише текстом.

Можна використовувати f-рядки для зручного виведення результатів.

Завдання на обчислення

Завдання на обчислення:

використовуються змінні,
виконується арифметичне обчислення або перетворення,
можна використовувати input(), print(), int(), float().
без умовних вказівок (if, else, elif).

1. Периметр квадрата

Введи довжину сторони квадрата → обчисли та виведи периметр ($P = 4 * a$).

2. Площа прямокутника

Введи довжину і ширину → обчисли площу ($S = a * b$).

3. Об'єм куба

Введи довжину ребра → обчисли об'єм ($V = a^3$).

4. Довжина кола

Введи радіус → обчисли довжину кола ($L = 2 * \pi * r$, $\pi \approx 3.14159$).

5. Площа кола

Введи радіус → обчисли площу ($S = \pi * r^2$).

6. Середнє арифметичне трьох чисел

Введи три числа → обчисли їх середнє $((a + b + c) / 3)$.

7. Перетворення метрів у сантиметри

Введи відстань у метрах → виведи у сантиметрах ($cm = m * 100$).

8. Години → секунди

Введи кількість годин → виведи еквівалент у секундах ($год * 3600$).

9. Ціна з ПДВ

Введи ціну без ПДВ → обчисли ціну з ПДВ (20%) → $ціна * 1.2$.

10. Знижка 10%

Введи початкову ціну → обчисли ціну зі знижкою 10% → $ціна * 0.9$.

11. Швидкість

Введи відстань (км) і час (год) → обчисли швидкість ($v = s / t$).

12. Потужність

Введи роботу (Дж) і час (с) → обчисли потужність ($P = A / t$).

13. Гіпотенуза прямокутного трикутника

Введи два катети a, b → обчисли гіпотенузу ($c = \sqrt{a^2 + b^2}$, використай $**0.5$).

14. Периметр трикутника

Введи три сторони → обчисли периметр ($P = a + b + c$).

15. Об'єм циліндра

Введи радіус основи і висоту → обчисли об'єм ($V = \pi * r^2 * h$).

16. Кінетична енергія

Введи масу (кг) і швидкість (м/с) → обчисли енергію ($E = 0.5 * m * v^2$).

17. Фаренгейти → Цельсії

Введи температуру у °F → обчисли в °C: $(F - 32) * 5 / 9$.

18. Кількість днів у секундах

Введи кількість днів → виведи у секундах (дні * 24 * 3600).

19. Подвійна сума двох чисел

Введи два числа → обчисли подвоєну суму ($2 * (a + b)$).

20. Обернене число

Введи число x (вважай, що $x \neq 0$) → обчисли $1 / x$.

Ці програми:

розв'язують конкретні обчислювальні задачі,
підходять для практики введення, змінних, арифметики та виведення
результатів.

Можна реалізувати кожну за 3–5 рядків коду.

Більш складніші обчислення

Завдання на складніші обчислення:

потрібні формули надаються,
без умовних вказівок (if, else, elif),
без тригонометричних функцій (sin, cos, tan тощо),
використовуються змінні, input(), print(), арифметичні операції, степені, корені,
логіка обчислень.

1. Середня швидкість руху на двох ділянках

Введи довжини двох ділянок (s_1 , s_2) і час на кожній (t_1 , t_2).

Обчисли середню швидкість: $(s_1 + s_2) / (t_1 + t_2)$.

2. Опір двох паралельних резисторів

Введи опори R_1 , R_2 .

Обчисли: $R = 1 / (1/R_1 + 1/R_2)$.

3. Гармонійне середнє двох чисел

Введи два додатні числа a , b .

Обчисли: $H = 2 * a * b / (a + b)$.

4. Об'єм конуса

Введи радіус основи r і висоту h .

Обчисли: $V = (1/3) * \pi * r^2 * h$ ($\pi \approx 3.14159$).

5. Площа поверхні кулі

Введи радіус r .

Обчисли: $S = 4 * \pi * r^2$.

6. Гравітаційна сила між двома тілами

Введи маси m_1 , m_2 і відстань r .

Обчисли: $F = G * m_1 * m_2 / r^2$, де $G = 6.67430e-11$.

7. Час падіння тіла (без опору повітря)

Введи висоту h (у метрах).

Обчисли: $t = \sqrt{2 * h / g}$, де $g = 9.81$.

8. Швидкість після прискорення

Введи початкову швидкість v_0 , прискорення a , час t .

Обчисли кінцеву швидкість: $v = v_0 + a * t$.

9. Пройдений шлях при рівноприскореному русі

Введи v_0 , a , t .

Обчисли: $s = v_0 * t + 0.5 * a * t^2$.

10. Енергія фотона за частотою

Введи частоту ν (Гц).

Обчисли: $E = h * \nu$, де $h = 6.626e-34$.

11. Довжина кола описаного навколо квадрата

Введи сторону квадрата a .

Радіус описаного кола: $R = a / \sqrt{2}$.

Довжина: $L = 2 * \pi * R$.

12. Площа трапеції

Введи основи a , b і висоту h .

Обчисли: $S = (a + b) / 2 * h$.

13. Кількість калорій при бігу

Введи вагу (кг) і відстань (км).

Приблизно: $\text{кал} = \text{вага} * \text{відстань} * 1.036$.

14. Індекс маси тіла (ІМТ)

Введи масу (кг) і зріст (м).

Обчисли: $BMI = \text{маса} / (\text{зріст} ** 2)$.

15. Ємність двох послідовних конденсаторів

Введи C_1 , C_2 .

Обчисли: $C = 1 / (1/C_1 + 1/C_2)$.

16. Площа правильного шестикутника за стороною

Введи сторону a .

Обчисли: $S = (3 * \sqrt{3} / 2) * a^2 \rightarrow$ використай $3**0.5$ для $\sqrt{3}$.

17. **Час подорожі з урахуванням середньої швидкості**
Введи відстань s і середню швидкість v .
Обчисли: $t = s / v$.
18. **Кінетична енергія обертання**
Введи момент інерції I і кутову швидкість ω .
Обчисли: $E = 0.5 * I * \omega^2$.
19. **Формула Герона (площа трикутника за трьома сторонами)**
Введи сторони a , b , c .
Півпериметр: $p = (a + b + c) / 2$.
Площа: $S = \sqrt{(p * (p - a) * (p - b) * (p - c))}$.
20. **Робота електричного струму**
Введи напругу U , силу струму I , час t (у секундах).
Обчисли: $A = U * I * t$.

Усі завдання:

не вимагають умов,
не використовують тригонометрію,
базуються на фізичних, геометричних або практичних формулах,
розвивають навички роботи з дробами, степенями, коренями, константами.

Можна реалізувати кожну програму за 4–7 рядків коду.

Виконання вказівок за умовою з використанням if (без else)

Завдання, у яких використовується умовна вказівка if без else (тобто виконується дія лише за умови істинності). Усі завдання передбачають введення даних через input() і використання числових або рядкових порівнянь.

1. Запитай вік користувача. Якщо вік ≥ 18 , виведи: "Ти повнолітній".
2. Запитай число. Якщо число > 0 , виведи: "Число додатне".
3. Запитай ім'я. Якщо ім'я == "Адмін", виведи: "Привіт, Адмін!".
4. Запитай температуру (число). Якщо температура < 0 , виведи: "На вулиці мороз!".
5. Запитай кількість балів. Якщо бали ≥ 90 , виведи: "Відмінно!".
6. Запитай слово. Якщо слово == "так", виведи: "Погоджено".
7. Запитай швидкість автомобіля. Якщо швидкість > 60 , виведи: "Перевищення швидкості!".
8. Запитай число. Якщо число $\% 2 == 0$, виведи: "Число парне".
9. Запитай назву міста. Якщо місто == "Київ", виведи: "Столиця України!".
10. Запитай ціну товару. Якщо ціна > 1000 , виведи: "Це дорогий товар".

11. Запитай кількість товару в наявності. Якщо кількість == 0, виведи: "Товару немає в наявності".
12. Запитай рік народження. Якщо рік < 2000, виведи: "Ти народився(лась) у минулому столітті".
13. Запитай оцінку (від 1 до 12). Якщо оцінка ≥ 10, виведи: "Високий рівень!".
14. Запитай рядок. Якщо рядок == "", виведи: "Ви нічого не ввели".
15. Запитай вагу в кг. Якщо вага > 100, виведи: "Вага перевищує 100 кг".
16. Запитай назву мови програмування. Якщо вона == "Python", виведи: "Правильний вибір!".
17. Запитай кількість секунд. Якщо секунд ≥ 60, виведи: "Це принаймні одна хвилина".
18. Запитай пароль. Якщо пароль == "12345", виведи: "Доступ дозволено".
19. Запитай довжину сторони квадрата. Якщо сторона > 10, виведи: "Квадрат великий".
20. Запитай кількість помилок у коді. Як Запитай кількість помилок у коді. Якщо помилок == 0, виведи: "Код без помилок!".

Усі завдання використовують тільки if, без else чи elif. Це допомагає зосередитися на логіці умови та дії, яка виконується лише за її істинності. Завдання охоплюють як числові, так і рядкові порівняння.

Виконання за умовою з використанням if-else (без elif)

Завдання, у яких використовується **умовні вказівки if-else** (без elif). Кожна програма приймає вхідні дані через input(), виконує перевірку умови та виводить один з двох можливих результатів.

1. **Парне чи непарне**
Введи ціле число. Якщо воно парне — виведи "Парне", інакше — "Непарне".
2. **Повноліття**
Введи свій вік. Якщо ≥ 18 — "Ти повнолітній", інакше — "Ти неповнолітній".
3. **Додатне чи від'ємне**
Введи число. Якщо > 0 — "Додатне", інакше — "Недодатне".
4. **Чи порожній рядок?**
Введи рядок. Якщо рядок == "" — "Порожній", інакше — "Не порожній".
5. **Оцінка за 12-бальною шкалою**
Введи оцінку (1–12). Якщо ≥ 7 — "Зараховано", інакше — "Не зараховано".
6. **Чи ділиться на 5?**
Введи ціле число. Якщо ділиться на 5 (остача 0) — "Ділиться", інакше — "Не ділиться".

7. Температура: тепло чи холодно?

Введи температуру. Якщо ≥ 20 — "Тепло", інакше — "Холодно".

8. Чи співпадають імена?

Введи два імені. Якщо однакові — "Однакові", інакше — "Різні".

9. Чи є літера голосною?

Введи одну літеру (у нижньому регістрі, малу). Якщо це 'a', 'e', 'i', 'o', 'u' — "Голосна", інакше — "Приголосна".

10. Чи достатньо грошей?

Введи суму грошей і ціну товару. Якщо грошей $\geq$ ціни — "Достатньо", інакше — "Недостатньо".

11. Перевірка паролю

Введи пароль. Якщо він == "secret" — "Доступ дозволено", інакше — "Доступ заборонено".

12. Чи більше 100?

Введи число. Якщо > 100 — "Більше сотні", інакше — "Сотня або менше".

13. Чи є рік високосним? (спрощено: ділиться на 4)

Введи рік. Якщо $\text{рік} \% 4 == 0$ — "Високосний", інакше — "Звичайний".

14. Чи входить число в діапазон [10, 50]?

Введи число. Якщо $10 \leq \text{число} \leq 50$ — "У діапазоні", інакше — "Поза діапазоном".

15. Чи є символ цифрою?

Введи один символ. Якщо `символ.isdigit()` — "Цифра", інакше — "Не цифра".

16. Чи дорівнює сума цифр 10?

Введи двозначне число. Якщо сума його цифр == 10 — "Сума 10", інакше — "Сума не 10".

17. Порівняння двох чисел

Введи два числа. Якщо перше $>$ другого — "Перше більше", інакше — "Перше не більше".

18. Чи є слово довгим?

Введи слово. Якщо довжина ≥ 8 — "Довге слово", інакше — "Коротке слово".

19. Чи ввів користувач "так"?

Введи слово. Якщо у нижньому регістрі == "так" — "Підтверджено", інакше — "Не підтверджено".

20. Чи є число нульовим?

Введи число. Якщо == 0 — "Нуль", інакше — "Не нуль".

Усі програми:

використовують тільки одне розгалуження один if-else,
не містять elif,
охоплюють числові, рядкові, логічні та практичні сценарії.

Кожну можна реалізувати за 3–6 рядків коду.

Повна умовна конструкція if-elif-else

Завдання, у яких використовується повна умовна конструкція **if-elif-else**. Кожне завдання передбачає три або більше можливих результатів на основі введених даних.

1. Оцінка за 12-бальною шкалою

Введи оцінку (1–12):

- 10–12 → "Відмінно"
- 7–9 → "Добре"
- 4–6 → "Задовільно"
- інакше → "Незадовільно"

2. Вікова категорія

Введи вік:

- < 6 → "Дошкільник"
- 6–17 → "Учень"
- 18–60 → "Дорослий"
- 60 → "Пенсіонер"

3. Температура повітря

Введи температуру (°C):

- ≥ 30 → "Спека"
- 15–29 → "Тепло"
- 0–14 → "Прохолодно"
- < 0 → "Мороз"

4. Швидкість автомобіля

Введи швидкість (км/год):

- 120 → "Дуже швидко!"
- 60–120 → "Нормально"
- < 60 → "Повільно"

5. Довжина слова

Введи слово:

- ≤ 3 літери → "Коротке"
- 4–7 літер → "Середнє"
- ≥ 8 літер → "Довге"

6. Кількість балів на тесті (0–100)

- ≥ 90 → "A"
- 80–89 → "B"
- 70–79 → "C"
- 60–69 → "D"
- < 60 → "F"

7. Вибір мови (за кодом)

Введи код: "uk", "en", "de", інше:

"uk" → "Українська"

"en" → "English"

"de" → "Deutsch"

інакше → "Невідома мова"

8. Розмір одягу за зростом (см)

< 150 → "Дитячий"

150–170 → "S"

171–185 → "M"

185 → "L"

9. Кількість днів у місяці (за номером 1–12)

Вважай, що лютого — 28 днів:

1,3,5,7,8,10,12 → 31

4,6,9,11 → 30

2 → 28

інакше → "Невірний номер місяця"

10. Індекс маси тіла (ІМТ)

Введи масу (кг) і зріст (м), обчисли $IMT = \text{маса} / \text{зріст}^2$:

< 18.5 → "Недовага"

18.5–24.9 → "Норма"

25–29.9 → "Надлишкова вага"

≥ 30 → "Ожиріння"

11. Час доби за годиною (0–23)

6–11 → "Ранок"

12–17 → "День"

18–23 → "Вечір"

0–5 → "Ніч"

12. Рівень знань мови програмування

Введи кількість місяців досвіду:

0 → "Ніколи не вчив"

1–6 → "Початківець"

7–18 → "Середній рівень"

18 → "Досвідчений"

13. Тип трикутника за кутами

Введи один кут (вважаємо прямокутний, гострокутний, тупокутний):

== 90 → "Прямокутний"

< 90 → "Гострокутний"

90 → "Тупокутний"

14. Стан води за температурою (°C)

≤ 0 → "Лід"

1–99 → "Вода"

- $\geq 100 \rightarrow$ "Пара"
15. **Рівень гучності (дБ)**
Введи рівень шуму:
 $< 40 \rightarrow$ "Тихо"
 $40-70 \rightarrow$ "Помірно"
 $70 \rightarrow$ "Гучно"
16. **Категорія ваги поштового листа (грами)**
 $\leq 20 \rightarrow$ "Лист"
 $21-100 \rightarrow$ "Невеликий пакет"
 $101-500 \rightarrow$ "Пакет"
 $500 \rightarrow$ "Велика посилка"
17. **Сезон за місяцем (1–12)**
 $12,1,2 \rightarrow$ "Зима"
 $3,4,5 \rightarrow$ "Весна"
 $6,7,8 \rightarrow$ "Літо"
 $9,10,11 \rightarrow$ "Осінь"
інакше $\rightarrow$ "Невірний місяць"
18. **Рівень заряду батареї (%)**
 $\geq 80 \rightarrow$ "Повний заряд"
 $30-79 \rightarrow$ "Середній заряд"
 $< 30 \rightarrow$ "Низький заряд"
19. **Тип числа**
Введи ціле число:
 0 і парне $\rightarrow$ "Додатне парне"
 0 і непарне $\rightarrow$ "Додатне непарне"
 $< 0 \rightarrow$ "Від'ємне"
 $== 0 \rightarrow$ "Нуль"
20. **Вибір режиму роботи (введи: 'відпочинок', 'робота', 'навчання')**
"робота" $\rightarrow$ "Час працювати!"
"навчання" $\rightarrow$ "Час вчитися!"
"відпочинок" $\rightarrow$ "Відпочивай!"
інше $\rightarrow$ "Невідомий режим"

Ці завдання:

охоплюють числові, часові, текстові та практичні ситуації,
вимагають використання мінімум одного if, одного або кількох elif та одного else,
добре підходять для практики логічного розгалуження в Python.

Кожне можна реалізувати за 5–10 рядків коду

Цикли з передумовою

Завдання, у яких використовується **цикл while**. Усі вони передбачають повторення дій, поки виконується умова, і не використовують **for**. Деякі завдання можуть включати введення даних, лічильники, накопичення суми/кількості, або інтерактивні цикли.

1. Лічильник від 1 до N

Введи число N. Виводь усі цілі числа від 1 до N (включно), кожне в окремому рядку.

2. Зворотний відлік

Введи число N. Виводь усі числа від N до 1 (включно) у порядку спадання.

3. Сума чисел від 1 до N

Введи N. Обчисли та виведи суму всіх цілих чисел від 1 до N.

4. Введення, доки не введено "стоп"

Повторюй введення слів, поки користувач не введе слово "стоп". Після цього виведи "Кінець".

5. Лічильник парних чисел

Введи N. Підрахуй і виведи, скільки парних чисел у діапазоні від 1 до N.

6. Множення без оператора *

Введи два цілих додатних числа a і b. Обчисли їх добуток, додаючи a до себе b разів.

7. Степінь числа (без **)

Введи основу a і показник n (ціле ≥ 0). Обчисли a^n за допомогою множень у циклі.

8. Перевірка на просте число

Введи число $n > 1$. Визнач і виведи, чи є воно простим (використовуй **while** для ділення).

9. Введення додатного числа

Повторюй запит числа, доки користувач не введе додатне число. Потім виведи його.

10. Факторіал числа

Введи ціле $n \geq 0$. Обчисли та виведи $n!$.

11. Сума цифр числа

Введи ціле додатне число. Обчисли суму його цифр, використовуючи ділення на 10 у циклі.

12. Кількість цифр у числі

Введи ціле додатне число. Визнач і виведи, скільки цифр у ньому.

13. НСД двох чисел (алгоритм Евкліда)

Введи два додатні цілі числа. Обчисли їх найбільший спільний дільник за допомогою **while**.

14. Виведення символів у рядку по одному

Введи рядок. Виводь кожен символ окремо, використовуючи індекс і `while` (без `for`).

15. Гра "вгадай число" (з фіксованим загаданим)

Загадай число (наприклад, 7). Дозволь користувачу вводити числа, поки не вгадає. Виводь "Мало" або "Багато" після кожної спроби.

16. Перетворення числа в двійковий запис

Введи додатне ціле число. Виведи його двійкове представлення (без використання `bin()`), отримане через ділення на 2.

17. Обчислення середнього арифметичного

Повторюй введення чисел, поки не буде введено 0. Після цього виведи середнє арифметичне (крім нуля).

18. Пошук найбільшої цифри у числі

Введи додатне ціле число. Знайди та виведи найбільшу цифру у ньому.

19. Виведення степенів двійки $\leq N$

Введи N . Виводь усі числа виду $2^0, 2^1, 2^2, \dots$, які $\leq N$.

20. Перевірка на паліндром (числовий)

Введи ціле додатне число. Визнач, чи є воно паліндромом (читається однаково зліва направо і навпаки), використовуючи `while` для "перевороту" числа.

Ці завдання:

охоплюють лічильники, накопичення, обробку цифр, інтерактивні цикли, не використовують `for`, добре розвивають розуміння умови продовження циклу та оновлення змінних.

Кожне завдання можна виконати за 5–12 рядків коду.

Цикл `while` з вказівкою `break`

Завдання, у яких використовується цикл `while` разом із вказівкою `break`. У всіх випадках цикл має умову, що завжди істинна (наприклад, `while True:`), а вихід здійснюється через `break` за певної умови всередині циклу.

1. Введення, доки не введено "вихід"

Повторюй ввід слів. Якщо користувач вводить "вихід", заверши цикл і виведи "До побачення!".

2. Сума чисел до нуля

Вводь числа, доки не буде введено 0. Виведи суму всіх введених чисел (нуль не враховується).

3. Пошук першого від'ємного числа

Вводь числа, доки не зустрінеться перше від'ємне. Виведи: "Знайдено від'ємне: [число]".

4. **Введення додатного числа (з повідомленням про помилку)**
Вводь число. Якщо воно ≤ 0 — виводь "Спробуй ще раз". Якщо > 0 — виведи його і заверши.
5. **Гра: вгадай число (з обмеженням спроб)**
Загадай число (наприклад, 42). Дозволь не більше 5 спроб. Якщо вгадано — виведи "Вітаю!" і break. Після 5 спроб — "Спроби закінчилися".
6. **Зупинка за спеціальним кодом**
Вводь коди товарів. Якщо введено "STOP" — припини ввід і виведи "Ввід завершено".
7. **Перший паліндром**
Вводь слова. Як тільки зустрінеться слово-паліндром (однакове в обидва боки, наприклад, "око"), виведи його і заверши.
8. **Пошук числа, що ділиться на 7**
Вводь цілі числа. Як тільки зустрінеться число, кратне 7, виведи його і заверши.
9. **Ввід паролю з трьох спроб**
Дозволь ввести пароль до 3 разів. Якщо введено правильний ("1234") — "Успіх!" і break. Після трьох спроб — "Доступ заборонено".
10. **Збір ненульових чисел**
Вводь числа, доки не введеш 0. Після цього виведи "Нуль отримано — завершення".
11. **Пошук слова з літерою "х"**
Вводь слова. Як тільки зустрінеться слово, що містить літеру "х", виведи "Знайдено: [слово]" і заверши.
12. **Перший простий введений номер**
Вводь цілі числа, поки не зустрінеться просте число. Виведи його і зупинись.
13. **Ввід до порожнього рядка**
Вводь рядки. Якщо введено порожній рядок ("") — виведи "Порожній рядок — вихід" і заверши.
14. **Сума до досягнення межі**
Вводь додатні числа, доки їхня сума не стане ≥ 100 . Виведи остаточну суму.
15. **Введення унікальних імен**
Вводь імена. Якщо користувач вводить те саме ім'я двічі поспіль — виведи "Повтор!" і заверши.
16. **Пошук парного числа**
Вводь цілі числа. Як тільки зустрінеться парне — виведи "Парне: [число]" і зупинись.
17. **Завершення за командою "q"**
Вводь будь-які символи. Якщо введено "q" — виведи "Вихід за запитом" і заверши.
18. **Обробка до першого дробового числа**
Вводь числа як рядки. Як тільки введено число з крапкою (наприклад, "3.14") — перетвори на float, виведи його і заверши.

19. Перший нуль у послідовності

Вводь числа. Як тільки зустрінеться 0 — виведи "Нуль отримано на кроці [номер]" і заверши (використовуй лічильник).

20. Зупинка при введенні "готово" або "done"

Вводь слова. Якщо введено "готово" або "done" — виведи "Завершення роботи" і вийди з циклу.

Усі завдання:

використовують `while True`;
передбачають умову виходу всередині циклу через `break`,
розвивають навички контролю потоку виконання, обробки вводу та прийняття рішень у циклі.

Кожне завдання можна реалізувати за 5–10 рядків коду.

Цикли з відомою кількістю повторень

Завдання, у яких використовується цикл `for` з `range(n)`, де `range()` має лише один параметр (тобто початок — 0, крок — 1, кінець — n). Ці завдання не використовують інші форми `range()` (з двома чи трьома аргументами), а також не використовують `while`, `if` (або використовують лише там, де це не змінює логіку циклу, наприклад, для введення даних).

1. Виведення чисел від 0 до $N-1$

Введи число N . Виведи всі цілі числа від 0 до $N-1$, кожне в окремому рядку.

2. Сума чисел від 0 до $N-1$

Введи N . Обчисли та виведи суму всіх цілих чисел від 0 до $N-1$.

3. N разів — слово "Привіт"

Введи N . Виведи слово "Привіт" рівно N разів (кожне — з нового рядка).

4. Виведення квадратів чисел

Введи N . Для кожного i від 0 до $N-1$ виведи i^2 .

5. N зірочок у стовпчик

Введи N . Виведи N зірочок (*), кожен в окремому рядку.

6. Кількість ітерацій

Введи N . Виведи номер кожної ітерації (від 1 до N), використовуючи `range(N)`.

7. Виведення парних індексів

Введи N . Виведи всі парні числа серед 0, 1, ..., $N-1$ (використовуй умову всередині циклу, але `range` — з одним параметром).

8. Обчислення факторіалу через діапазон

Введи N (≥ 0). Обчисли $N!$, використовуючи `for i in range(N)`: (помнож на $i+1$).

9. N пустот

Введи N . Виведи N порожніх рядків (або просто `print()` N разів).

10. **Генерація послідовності "Крок 0", "Крок 1", ..., "Крок N-1"**
Введи N. Виведи N рядків у форматі: "Крок 0", "Крок 1", ..., "Крок {N-1}".
11. **Виведення символів алфавіту (перші N літер)**
Введи N ($1 \leq N \leq 26$). Виведи перші N літер латинського алфавіту (використовуй `chr(ord('a') + i)`).
12. **Підрахунок загальної кількості ітерацій**
Введи N. Просто виконай цикл N разів і в кінці виведи "Виконано N ітерацій".
13. **Виведення степенів двійки (2^0 , 2^1 , ..., 2^{N-1})**
Введи N. Для кожного i в `range(N)` виведи 2^{**i} .
14. **Прямокутник з зірочок (N рядків, по 5 зірочок)**
Введи N. Намалюй прямокутник: N рядків, у кожному — *****.
15. **Виведення чисел у зворотному порядку (через індекс)**
Введи N. Виведи числа від N-1 до 0, використовуючи `for i in range(N):` і обчислення N-1-i.
16. **Повторення рядка N разів**
Введи рядок s і число N. Виведи цей рядок N разів (кожен — з нового рядка).
17. **Виведення індексів від 0 до N-1 у одному рядку через пропуск**
Введи N. Виведи усі індекси (0 1 2 ... N-1) в одному рядку.
18. **N трикутників (кожен — з однієї зірочки)**
Введи N. Намалюй N "трикутників", кожен — як * в окремому рядку.
19. **Обчислення суми квадратів**
Введи N. Обчисли суму квадратів чисел від 0 до N-1.
20. **Повторне виведення номера ітерації (з початку від 1)**
Введи N. Для кожної ітерації виведи: "Ітерація: 1", "Ітерація: 2", ..., "Ітерація: N".

Усі завдання:

використовують лише `range(N)` (один аргумент),
не використовують `range(start, stop)` чи `range(start, stop, step)`,
можуть включати прості арифметичні дії, конкатенацію, виведення,
підходять для закріплення розуміння роботи `for` і `range`.

Кожне завдання реалізується за 3–7 рядків коду.

Різні параметри циклів з відомою кількістю повторень

Завдання на використання циклу `for` з функцією `range()`, де застосовуються різні форми `range()`:

`range(n)` (1 параметр),

`range(start, stop)` (2 параметри),
`range(start, stop, step)` (3 параметри).

Кожне завдання спеціально використовує одну з цих форм або поєднує їх для досягнення результату.

Завдання 1–7: `range(stop)` (1 параметр)

1. Виведи перші N натуральних чисел (від 1 до N), використовуючи `range(N)`.
2. Виведи квадрати чисел від 0 до $N-1$, використовуючи `range(N)`.
3. Намалюй сніжинку: виведи N рядків, кожен — з одного символу *, використовуючи `range(N)`.
4. Обчисли суму парних індексів у діапазоні $0..N-1$ (використовуй `range(N)` і умову `if i % 2 == 0`).
5. Виведи N разів слово "Python", використовуючи `range(N)`.
6. Створи список довжини N , заповнений нулями, використовуючи лише `range(N)` та `append()`.
7. Виведи номери ітерацій від 1 до N , використовуючи `range(N)` і додавання 1.

Завдання 8–13: `range(start, stop)` (2 параметри)

8. Виведи всі цілі числа від A до $B-1$ (включно A , не включаючи B).
9. Обчисли суму чисел від 10 до 20 (використай `range(10, 21)`).
10. Виведи парні числа від 2 до N (використай `range(2, N+1)` і крок 1, фільтруй через `if`).
11. Знайди кількість чисел у діапазоні $[L, R)$ — просто виведи $R - L$, але використай цикл `for i in range(L, R): count += 1`.
12. Виведи великі літери латиниці від 'C' до 'K', використовуючи `range(ord('C'), ord('K')+1)`.
13. Виведи всі непарні числа між M і N ($M < N$), використовуючи `range(M, N+1)` і `if`.

Завдання 14–20: `range(start, stop, step)` (3 параметри)

14. Виведи числа від 100 до 0 включно, крок -5 , тобто: 100, 95, 90, ..., 0.
15. Виведи всі парні числа від N до 2 включно (у зворотному порядку), використовуючи від'ємний крок.
16. Виведи кожне третє число від 1 до 30, тобто: 1, 4, 7, ..., 28.
17. Виведи степені двійки: $2^1, 2^2, \dots, 2^{10}$, використовуючи `range(1, 11)` — але обчислюй через цикл.
18. Намалюй лінію з N зірочок, але кожну другу заміни на +, використовуючи `range(0, N, 2)` для зміни.
19. Обчисли суму чисел, кратних 3, у діапазоні від 3 до 99, використовуючи `range(3, 100, 3)`.
20. Виведи числа від -10 до 10 з кроком 2: $-10, -8, -6, \dots, 8, 10$ — через `range(-10, 11, 2)`.

Примітки:

У завданнях 10, 13 тощо можна використовувати `if` для фільтрації, але **`range` має містити лише 2 аргументи**.

У завданнях 14–20 **обов'язково використовується 3-аргументний `range`**, часто з від'ємним або більшим за 1 кроком.

Усі завдання використовують **`for`**, а не `while`.

Завдання охоплюють **прямий і зворотний перебір, арифметичні прогресії, фільтрацію, рядки, суми**.

Ці завдання допоможуть глибоко зрозуміти гнучкість `range()` у поєднанні з `for`.

Вказівки `break` і `continue` в циклах

Завдання із використанням `break` і `continue` з циклами (`for` або `while`), змінними, числами, рядками та введенням через `input()`. Усі завдання враховують, що списки ще не вивчалися.

Завдання 1–10: з `continue` (без списків)

1. Пропуск непарних чисел

Використовуючи `for i in range(1, 21)`, виведи лише **парні** числа. Непарні — пропускай через `continue`.

2. Пропуск чисел, що діляться на 3

Виведи числа від 1 до 30, але **пропускай** ті, що діляться на 3, використовуючи `continue`.

3. Ігнорування малих літер

Введи один символ. Якщо це **мала літера** ('a'–'z'), пропусти її (`continue`), інакше — виведи.

4. Пропуск цифр у рядку

Введи рядок. Пройди по кожному символу (через `for char in рядок`) і **виводь лише ті, що не є цифрами** (використай `continue` для цифр).

5. Пропуск від'ємних чисел при введенні

Введи 5 чисел (по одному за раз). Для кожного: якщо число < 0 — пропусти його (`continue`), інакше — виведи.

6. Пропуск пропусків у введеному реченні

Введи речення. Виведи всі символи по одному, але **пропускай пропуски** за допомогою `continue`.

7. Лічильник без чисел, кратних 4

Використовуючи `for i in range(1, 51)`, підрахуй, скільки чисел **не діляться на 4**. Використай `continue` для кратних 4.

8. Пропуск символів, що не є літерами

Введи рядок. Виведи лише **літери** (латиниця або кирилиця — на вибір), пропускаючи все інше (continue).

9. Введення, доки не буде додатне число

Використовуй while True. Якщо введено ≤ 0 — continue, інакше — виведи число і break.

10. Пропуск голосних у слові

Введи слово. Виведи лише **приголосні** літери, пропускаючи голосні (а, е, і, о, у, я, ю, є, і тощо) через continue.

Завдання 11–20: з break (без списків)

11. Перше число ≥ 100

Введи числа в циклі (while True). Як тільки введеш число ≥ 100 — виведи його і break.

12. Зупинка при введенні "готово"

Вводь слова. Як тільки введеш "готово" — виведи "Дякую!" і заверши цикл (break).

13. Пошук першого парного числа

Вводь цілі числа (while True). Як тільки введеш **парне** — виведи "Парне: [число]" і break.

14. Введення додатного числа (з break)

Повторюй ввід, доки не отримаєш число > 0 . Тоді виведи його і break.

15. Перший символ 'х' у слові

Введи слово. Пройди по символах (for char in word). Як тільки зустрінеш 'х' або 'Х' — виведи "Знайдено х!" і break.

16. Сума до першого нуля

Введи числа. Накопичуй суму. Як тільки введеш 0 — виведи суму і break.

17. Зупинка при введенні порожнього рядка

Вводь рядки. Як тільки введеш "" — виведи "Кінець" і break.

18. Перше число, що ділиться на 7

Використовуй for i in range(1, 101). Як тільки зустрінеш число, кратне 7 — виведи його і break.

19. Гра: вгадай число (1–10)

Загадай число 7. Вводь спроби. Якщо користувач вводить 7 — виведи "Вітаю!" і break.

20. Кількість введених слів до "стоп"

Вводь слова. Рахуй, скільки слів введено **до** слова "стоп". Коли воно введено — виведи кількість і break.

Особливості:

Немає списків — лише змінні, числа, рядки, цикли.

Використання `for char in` рядок дозволено — це перебір символів, а не робота зі списками.

Усі завдання підходять для початківців, що знають лише: змінні, `input()`, `print()`, цикли, умови, `break`, `continue`.

Вказівки `break` і `continue` в циклах(зі списками)

Завдання, у яких використовуються оператори `break` і `continue` у циклах (`for` або `while`). У кожному завданні чітко вказано, коли слід **перервати** цикл (`break`) або **пропустити** ітерацію (`continue`). Завдання охоплюють числові, текстові та практичні сценарії.

Завдання 1–10: з використанням `continue`

1. Пропуск непарних чисел

Виведи всі **парні** числа від 1 до 20. Використай `for` і `continue`, щоб пропустити непарні.

2. Пропуск від'ємних чисел

Маєш список: [5, -3, 8, -1, 0, 12]. Виведи лише **невід'ємні** числа, використовуючи `continue`.

3. Ігнорування пропусків у рядку

Введи рядок. Виведи всі символи по одному, але **пропускай пропуски** за допомогою `continue`.

4. Пропуск кратних 5

Виведи числа від 1 до 30, але **не виводь числа, що діляться на 5**. Використай `continue`.

5. Пропуск коротких слів

Маєш список слів: ["я", "кіт", "у", "вікно"]. Виведи лише слова довжиною ≥ 3 , використовуючи `continue`.

6. Пропуск нульових значень

Введи 5 чисел. Виведи лише **ненульові**, пропускаючи нулі через `continue`.

7. Виведення лише голосних

Введи рядок. Пройди по всіх символах і виведи **лише голосні** (а, е, і, о, у), інші — пропускай (`continue`).

8. Пропуск парних індексів

Маєш список [10, 20, 30, 40, 50]. Виведи елементи з **непарних індексів** (1, 3, ...), використовуючи `continue`.

9. Пропуск чисел більших за 100

Введи числа, доки не введеш 0. Виводь лише ті, що ≤ 100 , пропускаючи інші (continue).

10. Пропуск символів, що не є цифрами

Введи рядок. Виведи лише цифри з нього, пропускаючи все інше (continue + char.isdigit()).

Завдання 11–20: з використанням break

11. Перше число, більше за 50

Маєш список: [12, 34, 55, 23, 89]. Знайди і виведи **перше число > 50**, потім зупинись (break).

12. Пошук слова "стоп" у списку

Маєш список слів. Проходь по ньому, і як тільки зустрінеш "стоп" — виведи "Зупинено" і заверши цикл.

13. Перший дільник числа (крім 1)

Введи число $n > 1$. Знайди і виведи **найменший дільник > 1**, потім зупинись (break).

14. Зупинка при введенні "exit"

Вводь слова в циклі. Як тільки введеш "exit" — вийди з циклу (break).

15. Перший від'ємний елемент

Маєш список чисел. Виведи **перше від'ємне число** і зупинись.

16. Сума до першого нуля

Введи числа в циклі. Накопичуй суму, але **зупинись, як тільки введеш 0** (нуль не додається).

17. Пошук паліндрома в списку слів

Маєш список слів. Знайди **перше слово-паліндром** (наприклад, "око") і виведи його, потім break.

18. Перше число, кратне 7

Виводь числа від 1 до 100, але **зупинись**, як тільки зустрінеш перше, що ділиться на 7.

19. Гра: вгадай число (з break)

Загадай число (наприклад, 25). Дозволь користувачу вводити спроби. Якщо вгадав — виведи "Так!" і break.

20. Зупинка при досягненні межі суми

Вводь додатні числа. Накопичуй суму. Як тільки сума стане ≥ 100 — виведи її і зупинись (break).

Примітки:

У завданнях можна використовувати **for** або **while** — на вибір, але логіка має передбачати break або continue.

break — повністю завершує цикл.

continue — переходить до наступної ітерації, **не виконуючи решту тіла циклу**.

Завдання добре підходять для розуміння керування потоком у циклах.

Кожне завдання можна реалізувати за 5–10 рядків коду.

Вкладені цикли

Завдання на вкладені цикли (for y for, for y while тощо), які **не використовують списки**, а лише базові конструкції: змінні, print(), input(), range(), умови, арифметику та роботу з рядками.

Геометричні фігури та малюнки (завдання 1–8)

1. Квадрат із зірочок

Введи N. Намалюй квадрат з N рядків і N зірочок у кожному рядку.

2. Прямокутник із зірочок

Введи висоту H і ширину W. Намалюй прямокутник із зірочок розміром $H \times W$.

3. Прямокутний трикутник (ліворуч)

Введи N. Намалюй трикутник:

```
*  
**  
***  
...  
...
```

4. Прямокутний трикутник (праворуч)

Введи N. Намалюй:

```
*  
**  
***
```

5. Порожнистий квадрат

Введи N. Намалюй квадрат, де лише межі — зірочки, а середина — пропуски.

```
*****  
*   *  
*   *  
*   *  
*   *  
*****
```

6. Піраміда (рівнобедрений трикутник)

Введи N. Намалюй піраміду.

7. Таблиця множення (1–9)

Виведи класичну таблицю множення: кожен рядок — множення числа від 1 до 9 на числа 1–9.

8. Ромб із зірочок

Введи N (непарне). Намалюй ромб, що складається з центрального рядка з N зірочок і симетричних верхньої/нижньої частин.

Числові завдання (завдання 9–14)

9. Всі пари чисел (i, j) , де $i < j \leq N$

Введи N . Виведи всі пари (i, j) у форматі $i * j = \dots$ для $1 \leq i < j \leq N$.

10. Виведення квадратної таблиці чисел

Введи N . Виведи таблицю $N \times N$, де кожна клітинка — це номер стовпця (або рядка).

11. Сума всіх добутків $i \cdot j$ для i, j від 1 до N

Введи N . Обчисли загальну суму всіх добутків $i * j$, де i, j від 1 до N .

12. Виведення координат клітинок

Введи H і W . Виведи всі координати у форматі (рядок, стовець), наприклад: $(1,1), (1,2), \dots, (H,W)$.

13. Чи є число простим? (перевірка дільників)

Введи N . Використовуючи вкладений цикл, перевір всі можливі дільники і виведи "Просте" або "Складене".

14. Виведення степенів: a^b для a, b від 1 до 5

Виведи усі значення $a ** b$, де a і b від 1 до 5, у вигляді таблиці.

Робота з рядками (завдання 15–20)

15. Таблиця символів (латиниця)

Виведи таблицю 5×5 латинських літер, починаючи з 'a':

a b c d e

f g h i j

...

16. Виведення всіх пар символів у слові

Введи слово. Виведи всі можливі пари символів (i, j) , де i і j — індекси, $i \neq j$.

17. Порівняння двох слів посимвольно

Введи два слова однакової довжини. Виведи "співпадає" або "не співпадає" для кожного символу.

18. Квадрат із літер введеного слова

Введи слово довжини N . Намалюй квадрат $N \times N$, де кожен рядок — це слово.

19. Виведення "матриці" з цифр від 0 до 9

Виведи таблицю 10×10 , де кожна клітинка — це цифра, що дорівнює (рядок + стовець) % 10.

20. Аналіз симетрії слова (попарне порівняння)

Введи слово. Використовуючи вкладені цикли, перевір, чи кожна літера i дорівнює літері $N-1-i$. (Примітка: можна зупинитися після першої невідповідності.)

Важливо:

Усі завдання можна вирішити без списків — лише через `print()`, `range()`, змінні, умови.

У завданнях на малювання можна використовувати `end=""` у `print()`, щоб уникати перенесень рядка.

Вкладених циклів може бути більше двох, але в більшості випадків достатньо двох.

Випадкові числа

Завдання на вивчення випадкових чисел у Python, у яких:

дозволено умовні оператори (`if`),

дозволено цикли (`for`, `while`),

використовується тільки модуль `random` з базовими функціями.

Генерація та прості умови (завдання 1–8)

1. Випадкове число від 1 до 10

Згенеруйте число і виведіть його. Якщо воно парне — додайте " (парне)".

2. Чи випало число ≥ 50 ?

Згенеруйте ціле число від 1 до 100. Виведіть "Більше або рівне 50", якщо так, інакше — "Менше 50".

3. Випадковий вибір так/ні

Згенеруйте число 0 або 1. Якщо 1 — виведіть "Так", інакше — "Ні".

4. Чи кратне 7?

Згенеруйте число від 1 до 100. Виведіть "Кратне 7", якщо ділиться на 7 без остачі.

5. Випадкова оцінка (1–12)

Згенеруйте оцінку. Якщо ≥ 10 — "Відмінно", 7–9 — "Добре", інакше — "Погано".

6. Чи випало просте число?

Згенеруйте число від 2 до 30. Перевірте "вручну" (циклом), чи є воно простим, і виведіть результат.

7. Випадковий день тижня (числом)

Згенеруйте число від 1 до 7. Виведіть "Вихідний", якщо 6 або 7, інакше — "Робочий день".

8. Чи менше за середнє?

Згенеруйте число від 1 до 20. Якщо воно < 10.5 — виведіть "Нижче середнього".

Цикли + випадкові числа (завдання 9–15)

9. Підрахунок парних серед 10 випадкових

Згенеруйте 10 разів числа від 1 до 20. Підрахуйте, скільки з них парні.

10. Перший нуль

Генеруйте цілі числа від 0 до 5, доки не з'явиться 0. Виведіть, скільки спроб знадобилося.

11. Сума випадкових чисел до 100

Генеруйте числа від 1 до 10, додаючи їх до суми. Зупиніться, коли сума ≥ 100 . Виведіть кінцеву суму та кількість чисел.

12. Скільки разів випаде "6" при 20 кидках кубика?

Промодельуйте 20 кидків (числа 1–6). Підрахуйте, скільки разів випало 6.

13. Чек на "успіх"

Генеруйте числа від 1 до 100. "Успіх" — якщо число ≤ 10 . Повторюйте 50 разів і підрахуйте кількість успіхів.

14. До першого числа, кратного 5

Генеруйте числа від 1 до 30, доки не отримаєте кратне 5. Виведіть це число та номер спроби.

15. Максимум із 15 випадкових чисел

Згенеруйте 15 чисел у діапазоні 1–100. Знайдіть і виведіть найбільше (без списку — лише через змінну `max_val`).

Ігри та моделювання (завдання 16–20)

16. Гра: "Більше чи менше"

Загадайте випадкове число від 1 до 50. Користувач вводить своє. Виведіть "Більше", "Менше" або "Вгадав!".

17. Монетка: 10 підкидань

Промодельуйте 10 підкидань монетки (0 = решка, 1 = орел). Підрахуйте, скільки "орлів".

18. Чи буде хоча б один "успіх"?

У 20 спробах генеруйте число від 1 до 100. "Успіх" — якщо ≤ 5 . Якщо хоча б один успіх — виведіть "Є успіх!", інакше — "Немає".

19. Середнє арифметичне 30 випадкових чисел

Згенеруйте 30 чисел від 1 до 100. Обчисліть і виведіть їхнє середнє (сума / 30).

20. Чи зустрінеться число 13?

Згенеруйте 100 разів числа від 1 до 50. Якщо хоча б раз випало 13 — виведіть "Програш!", інакше — "Виграш!".

Підказки:

Для генерації цілих чисел використовуйте `random.randint(a, b)`.

Усі лічильники, суми, максимуми зберігайте в **змінних**, а не у списках.

Цикли `for i in range(N)`: — ідеальні для фіксованої кількості спроб.

Цикли while — для умовних повторень (наприклад, "доки не випаде 0").

Графіка з Turtle

Завдання для вивчення можливостей модуля turtle у Python. Усі завдання призначено для початківців і не вимагають знання списків, функцій (крім базового використання команд черепашки) або складних структур. Акцент — на керуванні черепашкою, малюванні фігур, використанні циклів та кольорів.

Перед виконанням кожного завдання в коді потрібно імпортувати модуль:

```
import turtle
```

Основні фігури (завдання 1–8)

1. Намалюй квадрат

Використай forward() і left() або right(), щоб намалювати квадрат зі стороною 100 пікселів.

2. Намалюй трикутник

Рівносторонній трикутник зі стороною 120 пікселів.

3. Намалюй прямокутник

Ширина — 150, висота — 80 пікселів.

4. Намалюй коло

Використай turtle.circle(50) або намалюй коло власноруч через багатокутник з великим числом сторін.

5. Намалюй п'ятикутник

Правильний п'ятикутник зі стороною 100.

6. Намалюй шестикутник

Правильний шестикутник (сторона — 80).

7. Зірка з 5 променів

Використай поворот на 144° після кожного кроку.

8. Декілька квадратів у ряд

Намалюй 4 квадрати (сторона 50), розташовані поруч, з проміжком 10 пікселів.

Кольори та стилі (завдання 9–12)

9. Квадрат червоного кольору

Встанови колір олівця на "red" і намалюй квадрат.

10. Трикутник з товстою лінією

Встанови товщину лінії (pensize(5)) і намалюй трикутник синього кольору.

11. Коло з заповненням

Намалюй жовте коло, заповнене червоним кольором (використай `begin_fill()` / `end_fill()`).

12. Райдужна зірка

Намалюй п'ятипроменеву зірку, де кожен промінь — іншого кольору (використай зміну кольору в циклі).

Цикли та повторення (завдання 13–16)

13. Спіраль з квадратів

Намалюй 36 квадратів, кожного разу повертаючи черепашку на 10° після квадрата.

14. Квітка з кіл

Намалюй 12 кіл радіусом 50, кожного разу повертаючи черепашку на 30° навколо центру.

15. Полотно з точок

Намалюй сітку з точок (використай `dot(10)`): 5 рядків, 5 стовпців, з кроком 30 пікселів.

16. Анімація: черепашка рухається і залишає слід

Намалюй лінію, де кожні 20 пікселів змінюється колір (червоний → зелений → синій → ...).

Керування положенням і орієнтацією (завдання 17–20)

17. Поверни черепашку в центр

Після малювання квадрата поверни черепашку точно в центр екрану (`turtle.goto(0, 0)`).

18. Підпиши фігуру

Намалюй трикутник, потім підніми олівець (`penup()`), перемістися нижче і напиши слово "Трикутник".

19. Дзеркальна фігура

Намалюй квадрат у верхньому лівому куті, потім — такий самий у нижньому правому (використай `goto()`).

20. Черепашка малює сонце

Намалюй велике жовте коло (сонце), а навколо — 12 променів (короткі лінії), розташовані через 30° .

Поради:

Використовуй `turtle.speed(0)` для миттєвого малювання (або 1–10 для анімації). Після завершення програми додай `turtle.done()` або `turtle.exitonclick()`, щоб вікно не закривалося.

Усі завдання можна виконати за 5–15 рядків коду.

Графіка у SimpleGraphics

Модуль SimpleGraphics є частиною загальнодоступних пакетів PyPI , його потрібно встановити за допомогою команди **pip install simplegraphics-python**. У ЄPython цей модуль доступний без додаткового встановлення.

SimpleGraphics — бібліотека, розроблена Беном Стефенсоном (автором підручника *"The Python Workbook"* та *"Introduction to Programming Using Python"*) для навчальних цілей. Ця бібліотека надає простий інтерфейс для малювання (подібно до turtle, але з вікном у стилі tkinter).

Основні функції SimpleGraphics (пам'ятка):

```
from SimpleGraphics import *
```

```
# Вікно створюється автоматично
```

```
line(x1, y1, x2, y2)
```

```
circle(x, y, r)
```

```
ellipse(x, y, w, h)
```

```
rect(x, y, w, h)
```

```
line(x1, y1, x2, y2)
```

```
text(x, y, "текст")
```

```
setOutline("red")    # колір контуру
```

```
setFill("blue")      # колір заповнення
```

```
setBackground("white")
```

Координати: верхній лівий кут — (0, 0); вісь Y спрямована **вниз**.

1. Червоний квадрат

Намалюйте квадрат зі стороною 100 пікселів, розташований у точці (50, 50), червоного контуру та без заливки.

2. Синє коло з жовтою заливкою

Намалюйте коло радіусом 60 у центрі вікна (припустимо, розмір вікна 800×600), з синім контуром і жовтою заливкою.

3. Прямокутник з написом

Намалюйте прямокутник (200×100) у верхньому лівому куті. Над ним виведіть текст "Меню".

4. Три кольорові еліпси

Намалюйте три еліпси (ширина 80, висота 50), розташовані горизонтально з проміжком. Заливка: червоний, зелений, синій.

5. Смайлик

Намалюйте смайлик: велике жовте коло (обличчя), два чорні кола (очі), дуга або крива (усмішка).

6. Діаграма

Намалюйте три стовпчики (прямокутники) різної висоти, що представляють значення 150, 200, 100.

7. Декартова система координат

Намалюйте осі X і Y (від центру екрану до країв), підпишіть їх "X" і "Y".

8. Трикутник за трьома точками

Намалюйте трикутник через три виклики `line()`, з'єднуючи точки (100,100), (200,100), (150,50).

9. Червоний хрест

Намалюйте хрест у центрі екрану: дві лінії (вертикальна і горизонтальна), завтовшки 10 пікселів (використайте `rect` для товстих ліній).

10. Мішень

Намалюйте мішень з трьох кіл (радіуси 30, 60, 90), чергуючи кольори заливки: червоний, білий, червоний.

11. Дві фігури з різними кольорами контуру

Намалюйте коло (радіус 40) з зеленим контуром і прямокутник (60×40) з фіолетовим контуром.

12. Текст у різних кольорах

Виведіть три рядки тексту: "Привіт", "Світ", "Python" — кожен іншого кольору (використайте `setColor()` або `setOutline()` перед `text()`).

13. Прапор України

Намалюйте прямокутник 300×200, розділений навпіл: верхня половина — синя, нижня — жовта.

14. Дорога з розміткою

Намалюйте чорний прямокутник (дорога) та білі прямокутники (пунктирна лінія) по центру.

15. Сніжинка з ліній

Намалюйте 6 ліній, що виходять із центру під кутами 0°, 60°, 120°, ..., 300°.

16. Шахівниця (4×4)

Намалюйте 16 квадратів (розмір 50×50), чергуючи чорний і білий колір заливки.

17. Сонце за горизонтом

Намалюйте горизонтальну лінію (земля), під нею — темно-зелений прямокутник, над нею — жовте коло (сонце).

18. Графік функції $y = x$

Намалюйте лінію від (100,100) до (500,500) і підпишіть її " $y = x$ ".

19. Кнопка "Старт"

Намалюйте прямокутник (120×40) із заокругленими кутами (можна імітувати) і текстом "Старт" по центру.

20. Дві фігури: одна поверх іншої

Намалюйте великий сірий прямокутник, а поверх нього — менше червоне коло. Переконайтеся, що порядок малювання правильний.

Використання PyGameZero

Завдання для використання PyGame Zero — легкої та дуже дружньої до початківців бібліотеки для створення ігор у Python.

Перед виконанням завдань розмістіть потрібні файли зображень та інших медіаресурсів у теках `images/` та `sounds/` вашого проекту:

```
your_game/
├── game.py          ← ваш основний файл
├── images/          ← сюди кладіть зображення (.png, .jpg)
│   ├── player.png
│   ├── enemy.png
│   └── background.jpg
└── sounds/          ← сюди кладуться звуки (.wav, .ogg)
    ├── jump.wav
    └── hit.ogg
```

Основи: Actor, фон, малювання (завдання 1–7)

1. Показати спрайт у центрі

Створи актора (наприклад, `player = Actor("alien")`) і намалюй його в центрі екрана у функції `draw()`.

2. Змінити розмір вікна та задній фон

Встанови розмір вікна на 800x600 і задній колір на блакитний (BLUE або (135, 206, 250)).

3. Два спрайти — один над іншим

Створи два актори (alien і heart). Розташуй їх вертикально по центру екрана, з проміжком.

4. Текст у верхньому лівому куті

Надрукуй напис "Рівень: 1" у верхньому лівому куті екрана у draw().

5. Спрайт із написом під ним

Намалюй актора, а під ним — ім'я (наприклад, "Гравець") з використанням screen.draw.text().

6. Зміна зображення актора

Створи актора, який спочатку використовує "alien", але після 3 секунд (використай лічильник у update()) змінює зображення на "alien_hurt".

7. Показ рахунку

Створи змінну score = 0. Виводь її у верхньому правому куті: "Очки: 0".



Управління клавішами (завдання 8–13)

8. Рух вліво/вправо

Зроби так, щоб актор рухався вліво/вправо при натисканні стрілок ← → зі швидкістю 5 пікселів.

9. Рух у чотири сторони

Додай можливість руху вгору/вниз (↑ ↓) до попереднього завдання.

10. Перемикання зброї клавішею "SPACE"

При натисканні SPACE змінюй зображення актора на "alien_with_gun" на 0.5 секунди, потім повертай "alien".

11. Анімація ходьби

При русі вліво/вправо змінюй зображення на "alien_walk1" / "alien_walk2" кожні 0.2 секунди.

12. Рух із обмеженням меж екрана

Заборони актору виходити за межі екрана (перевіряй actor.left, actor.right тощо).

13. Стрибок клавішею "UP"

При натисканні "UP" актор піднімається на 50 пікселів за 0.3 секунди, потім повертається.

Прості ігрові механіки (завдання 14–20)

14. Створення снаряду

При натисканні SPACE створюй снаряд (маленький актор, наприклад "bullet"), що з'являється біля гравця і рухається вправо.

15. Рух ворога

Створи ворога (enemy = Actor("enemy")), який рухається горизонтально від лівого краю до правого, потім — назад.

16. Зіткнення гравця з ворогом

Якщо player.colliderect(enemy), зміни задній колір на червоний і виведи "Поразка!".

17. Збір монет

Створи монету (`coin = Actor("coin")`). Якщо гравець торкається монети — збільш рахунок на 10 і перемісти монету в нове випадкове місце (`random.randint`).

18. Таймер зворотного відліку

Додай змінну `time_left = 30`. Кожну секунду зменшуй її на 1. Коли `time_left == 0` — виведи "Час вийшов!".

19. Життя гравця

Створи змінну `lives = 3`. При зіткненні з ворогом віднімай 1 життя. Якщо `lives == 0` — завершуй гру.

20. Простий рівень: дій дістатися до фінішу

Створи ціль (`goal = Actor("flag")`). Якщо гравець торкається цілі — виведи "Перемога!" і зупини всі рухи.

Робота з рядковими величинами

Завдання на роботу з рядковими величинами в Python, які охоплюють основні операції: конкатенацію, зрізи, перевірку символів, пошук, заміну, регістр, довжину тощо.

Умови:

Використовуються **лише рядки** (без списків, кортежів, регулярних виразів);
Дозволено використовувати вбудовані методи рядків (`len`, `upper`, `find`, `replace`, `isalpha`, `isdigit` тощо);
Бажано уникати імпорту додаткових модулів.

Основні операції (завдання 1–7)

1. Довжина рядка

Введи рядок. Виведи кількість символів у ньому.

2. Перший і останній символ

Введи непорожній рядок. Виведи перший і останній символ через кому.

3. Рядок у верхньому регістрі

Введи рядок. Виведи його всім великими літерами.

4. Перевірка на паліндром

Введи слово. Визнач, чи воно є паліндромом (читається однаково в обох напрямках, без урахування регістру).

5. Чи містить рядок цифри?

Введи рядок. Виведи "Містить цифри", якщо є хоча б одна цифра, інакше — "Лише літери та символи".

6. Зворотний рядок

Введи рядок. Виведи його у зворотному порядку (наприклад, "hello" → "olleh").

7. Повторення рядка

Введи рядок і число N. Виведи цей рядок, повторений N разів поспіль (без пропусків).

Пошук і заміна (завдання 8–13)

8. Скільки разів зустрічається літера?

Введи рядок і символ. Підрахуй, скільки разів символ зустрічається у рядку.

9. Перше входження підрядка

Введи рядок і слово. Виведи індекс початку першого входження слова, або -1, якщо його немає.

10. Заміна всіх пропусків на підкреслення

Введи речення. Заміни всі пропуски на _.

11. Чи починається з великої літери?

Введи слово. Виведи "Так", якщо перша літера — велика, інакше — "Ні".

12. Видалення всіх голосних

Введи слово. Виведи його без голосних літер (а, е, і, о, у, будь-якому регістрі).

13. Заміна префікса

Введи рядок, що починається з "не". Заміни цей префікс на "абсолютно не"(наприклад, "неможливо" → "абсолютно неможливо").



Зрізи та форматування (завдання 14–20)

14. Перші три символи

Введи рядок довжиною ≥ 3 . Виведи перші три символи.

15. Останні два символи

Введи рядок довжиною ≥ 2 . Виведи останні два символи.

16. Середній символ

Введи непарне слово (наприклад, довжина 5, 7 тощо). Виведи середній символ.

17. Рядок без першого і останнього символу

Введи рядок довжиною ≥ 2 . Виведи його без першого і останнього символу.

18. Чи однакові перші два і останні два символи?

Введи рядок довжиною ≥ 2 . Перевір, чи співпадають перші два і останні два символи (наприклад, "мама" → так).

19. "Дзеркальне" слово

Введи слово. Створи нове слово: перша половина + зворотна перша половина (наприклад, "абв" → "абвба").

20. Форматування імені

Введи ім'я (наприклад, "олена"). Перетвори на "Олена" (перша літера велика, решта — малі).

Підказки для реалізації:

Використовуй `s.lower()` для уніфікації регістру.

Для паліндрома: `s == s[::-1]`.

Для підрахунку символів: `s.count(char)`.

Для зрізів: `s[:3]`, `s[-2:]`, `s[1:-1]` тощо.

Перевір довжину перед зрізами, щоб уникнути помилок.

Створення, заповнення та виведення числових списків

Завдання на створення, заповнення та виведення числових списків у Python. Усі завдання передбачають використання **списків (list)**, але **не вимагають використання функцій, класів або модулів** (окрім базових можливостей Python). Упор зроблено на:

створення списку,

заповнення через `append()`, `input()`, або генерацію,

виведення елементів (у рядок, у стовпчик, з форматуванням).

Дозволено: `for`, `while`, `range()`, `input()`, `print()`, `len()`, арифметика.

Не використовувати: `numpy`, `map()`, спискові включення (якщо не вказано інше), функції (окрім базових).

Частина 1: Створення та заповнення списків (завдання 1–10)

1. Список з 5 нулів

Створи список, що містить п'ять нулів: `[0, 0, 0, 0, 0]`. Виведи його.

2. Заповнення списку вручну через ввід

Запитай у користувача 4 цілих числа. Додай їх до списку по черзі (через `append()`). Виведи список.

3. Список парних чисел від 2 до 20

Створи список `[2, 4, 6, ..., 20]`, використовуючи цикл `for` і `append()`.

4. Список квадратів чисел від 1 до N

Введи `N`. Створи список, де кожен елемент — квадрат числа від 1 до `N`.

5. Список випадкових чисел (імітація)

Без `random`! Створи список із 6 чисел: `[10, 25, 7, 42, 18, 33]`. Виведи його.

6. Список, заповнений однаковим числом

Введи число `x` і кількість `n`. Створи список із `n` копій числа `x` (наприклад: `x=5`, `n=3` → `[5, 5, 5]`).

7. Арифметична прогресія

Введи початкове значення a , крок d і кількість елементів n . Створи список: $[a, a+d, a+2d, \dots, a+(n-1)d]$.

8. Список цифр числа

Введи ціле додатне число (наприклад, 1234). Створи список його цифр: $[1, 2, 3, 4]$ (використай перетворення в рядок і `int()` для кожного символу).

9. Заповнення списку з кроком 0.5

Створи список $[0.0, 0.5, 1.0, 1.5, \dots, 5.0]$ (довжина 11 елементів), використовуючи цикл і додавання.

10. Список факторіалів

Введи N . Створи список, де елемент i — це $i!$ (для i від 0 до N). Нагадаємо: $0! = 1$.

Частина 2: Виведення списків (завдання 11–20)

11. Виведення списку у стовпчик

Маєш список чисел. Виведи кожен елемент у окремому рядку.

12. Виведення списку в один рядок через кому

Маєш список. Виведи його у форматі: 1, 2, 3, 4.

13. Виведення списку з індексами

Для кожного елемента списку виведи: "Елемент [індекс]: [значення]".

14. Виведення лише додатних чисел

Маєш список (наприклад, $[-3, 5, 0, -1, 8]$). Виведи тільки додатні елементи у рядок.

15. Виведення парних елементів з їхніми індексами

Пройди по списку. Якщо елемент парний — виведи: "Індекс $[i]$: [елемент]".

16. Виведення списку у зворотному порядку

Маєш список. Виведи його елементи від останнього до першого (не змінюючи сам список).

17. Сума списку і її виведення

Маєш список чисел. Обчисли суму і виведи: "Сума: [сума]".

18. Найбільший елемент і його індекс

Знайди найбільше число у списку і виведи: "Максимум: [значення] на позиції [індекс]".

19. Перетворення списку в рядок без пропусків

Маєш список цифр $[1, 2, 3]$. Виведи їх як одне число: "123" (використай конкатенацію рядків).

20. Графічне відображення списку (стовпчаста діаграма)

Маєш список невід'ємних цілих чисел (наприклад, $[3, 1, 4, 2]$). Для кожного числа виведи відповідну кількість зірочок:

*

**

Поради:

Для виведення в один рядок використовуй `print(..., end=" ")` або збирання рядка.

Для пошуку максимуму/суми можна використовувати прості цикли з лічильниками.

У завданні 19: `result = ""`; `for x in lst: result += str(x)`.

Пошук елемента у списку за певним критерієм

Завдання на пошук елемента у списку за певним критерієм.

Усі списки містять **елементи одного типу** (числа **або** рядки), але **не обов'язково одного розміру**. У завданнях потрібно знайти **індекс, значення** або **наявність** елемента, що задовольняє умову.

Використовуйте лише базові конструкції: `for`, `while`, умови, змінні. **Не використовуйте** `index()`, `in` (для пошуку), спискові включення або функції вищого порядку.

Числові списки (завдання 1–12)

1. Перший додатний елемент

Знайди **індекс першого додатного** числа у списку. Якщо немає — виведи -1.

2. Останнє парне число

Знайди **значення останнього парного** числа у списку. Якщо немає — виведи "Немає".

3. Найменший елемент

Знайди **індекс найменшого** елемента у списку.

4. Елемент, що дорівнює середньому арифметичному

Обчисли середнє арифметичне списку. Знайди **індекс першого елемента**, що дорівнює цьому середньому (якщо такий є).

5. Перший елемент, більший за задане число X

Введи число X. Знайди **індекс першого елемента**, що $> X$. Якщо немає — виведи -1.

6. Елемент із максимальною сумою цифр

Для кожного цілого числа (додатного) обчисли суму його цифр. Знайди **індекс числа з найбільшою сумою цифр**.

7. Перший нуль у списку

Знайди **індекс першого нуля**. Якщо немає — виведи "Нуля немає".

8. Число, що зустрічається рівно один раз

Знайди **перше число**, яке зустрічається у списку **лише один раз** (припустимо, таке є).

9. Елемент, найближчий до 100

Знайди **значення елемента**, що **найменше відрізняється від 100** (за модулем).

10. **Перший елемент, кратний 7**

Знайди індекс першого числа, що ділиться на 7 без остачі. Якщо немає — виведи -1.

11. **Найбільший від'ємний елемент**

Знайди значення найбільшого серед від'ємних чисел (найближчого до нуля). Якщо від'ємних немає — виведи "Немає від'ємних".

12. **Елемент, що дорівнює сумі сусідів**

Знайди індекс елемента (не крайнього), що дорівнює сумі попереднього і наступного. Якщо таких кілька — знайди перший.

Рядкові списки (завдання 13–20)

13. **Перший рядок, що починається з літери "A"**

Знайди індекс першого слова, що починається з великої "A" або малої "a". Якщо немає — виведи -1.

14. **Найдовше слово**

Знайди індекс найдовшого рядка у списку. Якщо їх кілька — знайди перший.

15. **Перше слово, що містить літеру "х"**

Знайди індекс першого слова, що містить літеру "х" (українську або латинську — на вибір). Якщо немає — виведи -1.

16. **Слово, що є паліндромом**

Знайди індекс першого слова, що є паліндромом (наприклад, "око", "радар"). Не чутливий до регістру.

17. **Перше слово з довжиною 5**

Знайди індекс першого слова, що містить рівно 5 літер.

18. **Слово, що складається лише з цифр**

Знайди індекс першого рядка, що повністю складається з цифр (використай `.isdigit()`).

19. **Перше слово, написане великими літерами**

Знайди індекс першого слова, де всі літери — великі (ігноруй рядки без літер).

20. **Слово із найбільшою кількістю голосних**

Визнач, у якому слові найбільше голосних (а, е, и, і, о, у, я, ю, є, ї). Виведи його індекс.

Рекомендації:

Проходь по списку за допомогою `for i in range(len(lst))`, щоб мати доступ і до індексу, і до значення (`lst[i]`).

Використовуй змінну-прапорець або змінну для зберігання кращого кандидата (для "най-" завдань).

Для рядків — використовуй `.lower()`, щоб уніфікувати регістр.

Усі завдання можна вирішити за **один прохід** по списку.

Обчислення підсумкових значень елементів списку

Завдання на обчислення підсумкових значень елементів списку: сума, добуток, середнє, кількість, максимум, мінімум тощо.

Усі завдання призначено для числових списків (цілих або дійсних чисел), використовують лише базові конструкції Python (цикл for/while, змінні, умови), **без використання вбудованих функцій** sum(), max(), min(), mean() тощо.

Сума та кількість (завдання 1–7)

1. **Сума всіх елементів списку**
Обчисли суму всіх чисел у списку.
2. **Сума додатних елементів**
Обчисли суму лише тих елементів, що > 0 .
3. **Кількість від'ємних чисел**
Порахуй, скільки елементів у списку < 0 .
4. **Сума елементів з парними індексами**
Обчисли суму елементів, що стоять на індексах 0, 2, 4,
5. **Сума модулів усіх елементів**
Обчисли суму $|a[i]|$ для кожного елемента списку.
6. **Кількість нулів у списку**
Порахуй, скільки разів зустрічається число 0.
7. **Сума елементів, що діляться на 3**
Обчисли суму всіх чисел, які **діляться на 3 без остачі**.

Добуток та інші операції (завдання 8–12)

8. **Добуток усіх елементів**
Обчисли добуток усіх чисел у списку. (Припустимо, список непорожній і не містить нуля, якщо не вказано інше.)
9. **Добуток додатних елементів**
Обчисли добуток лише **додатних** чисел. Якщо таких немає — виведи 1.
10. **Сума квадратів усіх елементів**
Обчисли суму $a[i]^2$ для кожного елемента.
11. **Сума перших N елементів**
Введи число N ($\leq$ довжини списку). Обчисли суму перших N елементів.
12. **Сума останніх K елементів**
Введи число K. Обчисли суму останніх K елементів списку.

Середнє, максимум, мінімум (завдання 13–20)

13. **Середнє арифметичне всіх елементів**
Обчисли середнє значення: сума / кількість.
14. **Середнє арифметичне додатних елементів**
Обчисли середнє лише серед **додатних** чисел. Якщо їх немає — виведи 0.
15. **Найбільший елемент у списку**
Знайди **максимальне** число (без використання `max()`).
16. **Найменший елемент у списку**
Знайди **мінімальне** число (без `min()`).
17. **Різниця між максимумом і мінімумом**
Обчисли `max - min`.
18. **Середнє арифметичне елементів з непарними індексами**
Обчисли середнє значення елементів на позиціях 1, 3, 5,
19. **Сума відхилень від середнього**
Обчисли середнє, потім суму (`a[i] - середнє`) для всіх `i`.
Очікуваний результат: **0** (перевірка розуміння).
20. **Кількість елементів, більших за середнє**
Обчисли середнє, потім порахуй, скільки елементів **> середнього**.

Поради:

Для суми ініціалізуй змінну як `total = 0`.

Для добутку — як `product = 1`.

Для максимуму — як `maximum` = перший елемент списку, потім оновлюй у циклі.

Для середнього — обчисли суму і поділи на `len(список)`.

Уважно обробляй випадки з **порожніми підмножинами** (наприклад, "додатні відсутні").

Сортування списків

Завдання на сортування (впорядкування) списків **без використання методу `.sort()` і функції `sorted()`**. Усі завдання передбачають **реалізацію власної логіки сортування** за допомогою циклів, умов і обміну елементів (наприклад, через алгоритми **сортування "бульбашкою"**, **вибір максимального/мінімального**, тощо).

Дозволено: `for`, `while`, змінні, індекси, обмін значень.

Заборонено: `list.sort()`, `sorted()`, `numpy`, інші готові функції сортування.

Сортування за зростанням (завдання 1–7)

1. **Сортування методом "бульбашки" (за зростанням)**
Відсортуй список цілих чисел за зростанням, використовуючи **бульбашкове сортування**.
2. **Сортування вибором мінімального елемента**
На кожному кроці шукай найменший елемент серед невідсортованої частини і міняй його місцями з поточним.
3. **Сортування вставками**
Відсортуй список, імітуючи вставку кожного елемента у вже відсортовану частину.
4. **Сортування за зростанням без зміни оригіналу (створи новий список)**
Створи новий список, заповнюючи його мінімальними елементами з вихідного (видаляючи з копії або позначаючи).
5. **Сортування за зростанням, але парні числа — спочатку**
Спочатку всі **парні** (відсортовані за зростанням), потім — **непарні** (теж за зростанням).
6. **Сортування за модулем (за зростанням |a|)**
Відсортуй список так, щоб спочатку йшли числа з найменшим модулем (наприклад: -1, 2, -3, 5).
7. **Сортування дробових чисел "вручну"**
Відсортуй список float-чисел за зростанням, використовуючи будь-який простий алгоритм.

Сортування за спаданням (завдання 8–12)

8. **Бульбашкове сортування за спаданням**
Відсортуй список так, щоб найбільші числа йшли спочатку.
9. **Сортування вибором максимального елемента**
На кожному кроці шукай найбільший елемент і став його в початок.
10. **Сортування за спаданням, але від'ємні — в кінці**
Спочатку додатні (за спаданням), потім — від'ємні (теж за спаданням).
11. **Сортування за останньою цифрою (за спаданням)**
Відсортуй числа так, щоб спочатку йшли ті, у яких остання цифра найбільша (наприклад: 29, 18, 7, 3 → 29 (9), 18 (8), 7 (7), 3 (3)).
12. **Сортування за спаданням без зміни оригінального списку**
Створи копію у новому списку, впорядковану за спаданням.

Складніші критерії (завдання 13–20)

13. **Сортування за кількістю цифр (за зростанням)**
Числа з меншою кількістю цифр — спочатку (наприклад: 5, 23, 100, 7 → 5, 7, 23, 100).
14. **Сортування: спочатку додатні за зростанням, потім від'ємні за спаданням**
Наприклад: [-5, 3, -1, 2] → [2, 3, -1, -5].

15. **Сортування за сумою цифр (за зростанням)**

Число 29 ($2+9=11$) йде після 38 ($3+8=11$)? → тоді за самим значенням. Але спочатку — за сумою.

16. **"Зигзагоподібне" сортування**

Спочатку мінімум, потім максимум, потім наступний мінімум, наступний максимум...

Наприклад: $[1, 2, 3, 4, 5] \rightarrow [1, 5, 2, 4, 3]$.

17. **Сортування за парністю індексу (елементи з парних індексів — спочатку)**

Відсортуй список так: спочатку всі елементи, що **були** на парних індексах (відсортовані за зростанням), потім — з непарних.

18. **Сортування з "обертанням" навколо нуля**

Відсортуй так: 0, 1, -1, 2, -2, 3, -3, ... (спочатку 0, потім пари за зростанням модуля).

19. **Сортування за кратністю числу X**

Введи число X. Спочатку йдуть числа, **кратні X** (за зростанням), потім — решта (теж за зростанням).

20. **Стабільне сортування за знаком**

Не змінюй відносний порядок додатних і від'ємних чисел, але перенеси всі додатні **на початок**, від'ємні — у кінець.

Наприклад: $[3, -1, 2, -4, 5] \rightarrow [3, 2, 5, -1, -4]$.

Поради:

Для **бульбашкового сортування**:

```
for i in range(len(lst)):
```

```
    for j in range(len(lst) - 1 - i):
```

```
        if lst[j] > lst[j + 1]:
```

```
            lst[j], lst[j + 1] = lst[j + 1], lst[j]
```

Для **вибору мінімального**: зберігай індекс мінімуму і міняй у кінці зовнішнього циклу.

Для завдань із **новим списком**: можна використовувати допоміжний список і видаляти мінімум з копії.

Випадкові величини та списки

Завдання на використання випадкових чисел у списках, які охоплюють:

заповнення списків випадковими значеннями,
аналіз (пошук, підрахунок, фільтрація),
перемішування,

вибір випадкових елементів.

Використовується модуль `random`. Усі завдання призначені для закріплення роботи з списками та випадковістю разом. Дозволено: `import random`, `list.append()`, `len()`, цикли, умови.

Основні функції: `randint()`, `choice()`, `choices()`, `sample()`, `shuffle()`.

Заповнення списків (завдання 1–6)

1. Список із 10 випадкових цілих чисел від 1 до 100

Створіть список, заповнений 10 випадковими числами з діапазону `[1, 100]`.

2. Список парних чисел

Заповніть список 8 випадковими **парними** числами від 2 до 50.

3. Список дробових чисел

Створіть список із 5 випадкових дійсних чисел у діапазоні `[0.0, 10.0]` (з точністю до 2 знаків).

4. Список із повторюваними значеннями

Заповніть список 12 елементами, кожен з яких — випадковий вибір із `[1, 2, 3]`.

5. Список унікальних чисел

Створіть список із 7 **різних** випадкових чисел від 1 до 50 (без повторень).

6. Список довжини N за вказівкою користувача

Введіть N. Заповніть список N випадковими числами від 0 до 9.

Аналіз списку (завдання 7–13)

7. Підрахунок кількості чисел > 50

У списку з 15 випадкових чисел (1–100) підрахуйте, скільки елементів > 50.

8. Знайти індекс першого числа, кратного 7

У списку випадкових чисел (1–100) знайдіть індекс першого елемента, що ділиться на 7.

9. Найчастіший елемент у списку

У списку із 20 випадкових чисел з `[1, 5]` знайдіть число, що зустрічається найчастіше.

10. Чи є в списку хоча б одне парне число?

Створіть список із 10 випадкових непарних чисел (1, 3, 5, ..., 19). Перевірте, чи є парне (очікуваний результат: ні).

11. Сума елементів на непарних позиціях

У списку з 12 випадкових чисел обчисліть суму елементів з індексами 1, 3, 5, ...

12. Чи всі числа у списку менші за 90?

Згенеруйте список із 10 чисел (1–100). Перевірте, чи **всі** < 90.

13. Максимальний елемент і його індекс

У списку з 15 випадкових чисел знайдіть найбільше значення та його індекс.

Перемішування, вибір, зміна (завдання 14–20)

14. Перемішати список

Створіть список [1, 2, 3, 4, 5]. Перемішайте його випадковим чином за допомогою `random.shuffle()`.

15. Випадковий вибір 3 елементів без повторень

Зі списку чисел від 1 до 20 виберіть 3 унікальні випадкові елементи (використайте `random.sample()`).

16. Випадковий вибір з повтореннями

Зі списку ['A', 'B', 'C'] виберіть 10 елементів з можливістю повторень (`random.choices()`).

17. Видалити випадковий елемент

Створіть список із 8 елементів. Видаліть один випадковий елемент (виберіть індекс через `randint`, потім `pop()`).

18. Вставити випадкове число в випадкову позицію

Маєте список з 5 чисел. Вставте нове випадкове число (1–100) у випадкову позицію.

19. Створити список із випадковими булевими значеннями

Заповніть список 10 елементами: кожен — True або False з ймовірністю 50/50.

20. "Генератор команд"

Маєте список із 12 імен. Випадковим чином розділіть їх на дві команди по 6 осіб (`random.shuffle` + зрізи).

Корисні підказки:

Для унікальних чисел: `random.sample(range(1, 51), 7)`

Для перемішування: `random.shuffle(my_list)` (змінює список **на місці**)

Для випадкового індексу: `idx = random.randint(0, len(lst)-1)`

Для дробових чисел: `round(random.uniform(0, 10), 2)`

Створення програм з графічним інтерфейсом

Завдання на використання модуля `tkinter` у Python — стандартної бібліотеки для створення графічних інтерфейсів користувача (GUI). Усі завдання призначені для початківців і не вимагають попередніх знань про класи або складні шаблони. Ці завдання охоплюють основні віджети (`Label`, `Button`, `Entry`, `Text`, `Checkbutton`, `Radiobutton`, `Listbox`, `Menu`) та ключові концепції (події, компонування, оновлення інтерфейсу).

Перед виконання переконайтеся, що використовуєте вказівки імпорту:
`from tkinter import *`

або:

```
tk. import tkinter as tk.
```

Основи: вікно та прості віджети (завдання 1–6)

1. Порожнє вікно

Створіть вікно з заголовком "Моє перше вікно" та розміром 300x200.

2. Кнопка "Привіт"

Додайте кнопку з написом "Привіт". При натисканні — нічого не робить (поки що).

3. Мітка з текстом

Додайте мітку (Label) з текстом "Вітаємо у Tkinter!" у центрі вікна.

4. Кнопка та мітка разом

Розташуйте кнопку зверху і мітку знизу у вікні (використайте pack()).

5. Поле вводу (Entry)

Додайте поле вводу, де користувач може ввести своє ім'я.

6. Кнопка "Очистити" для поля вводу

Додайте кнопку, яка очищає вміст поля вводу при натисканні.

Взаємодія: обробка подій (завдання 7–12)

7. Привітання за іменем

Після натискання кнопки "Сказати привіт" відображайте у мітці: "Привіт, [ім'я]!".

8. Підрахунок кліків

Створіть кнопку "Клацни мене". При кожному кліку збільшуйте лічильник і виводьте: "Кліків: 5".

9. Перевірка паролю

Поле вводу + кнопка "Увійти". Якщо введено "1234" — мітка: "Доступ дозволено", інакше — "Спробуй ще".

10. Подвійне поле вводу

Два поля: введи число → натисни кнопку → у другому полі з'являється подвоєне значення.

11. Перемикач кольору фону

Кнопка "Змінити колір". При натисканні фон вікна міняється між білим і синім.

12. Виведення тексту з поля у мітку

Користувач вводить текст → натискає кнопку → мітка показує цей текст.

Компонування та компоненти (завдання 13–17)

13. Калькулятор додавання

Два поля вводу (числа), кнопка "+", мітка для результату. Додайте два числа і покажіть суму.

14. Група радіокнопок

Додайте три радіокнопки: "Червоний", "Зелений", "Синій". Вибір змінює колір фону вікна.

15. Прапорці (Checkbutton)

Два прапорці: "Показувати привітання", "Показувати дату". Мітка оновлюється залежно від вибору.

16. Список (Listbox)

Додайте Listbox з трьома елементами: "Apple", "Banana", "Cherry". При натисканні на елемент — виводьте його у мітку.

17. Меню верхнього рівня

Створіть меню з пунктом "Файл" → підпункт "Вихід", який закриває вікно.

Міні-проекти (завдання 18–20)

18. Лічильник секунд (таймер)

Кнопка "Пуск" → кожну секунду оновлює мітку: "Секунд: 1", "Секунд: 2" тощо. (Використайте `after()` замість `time.sleep(!)`)

19. Простий блокнот

Велике текстове поле (Text), кнопка "Зберегти" — виводить вміст у консоль (або у мітку).

20. Вікно з інформацією про користувача

Поля: ім'я, вік, улюблений колір. Кнопка "Показати" → мітка: "Ім'я: Олена, Вік: 16, Колір: синій".

Поради:

Використовуйте `pack()` для простого компонування, `grid()` — для таблиць.

Для оновлення мітки: `label.config(text="новий текст")`.

Для читання з Entry: `entry.get()`.

Для обробки кліку: `button = tk.Button(text="...", command=ваша_функція)`.

Побудова діаграм і графіків

Завдання на побудову діаграм і графіків за допомогою `matplotlib` у Python. Усі завдання призначені для початківців і використовують лише основні функції: `plot()`, `bar()`, `pie()`, `xlabel()`, `title()` тощо.

Перед виконання потрібно імпортувати бібліотеку (`import matplotlib.pyplot as plt`).

Лінійні графіки (завдання 1–5)

1. Простий лінійний графік

Побудуйте графік функції $y = x$ для $x = [0, 1, 2, 3, 4, 5]$.

2. Графік квадратичної функції

Побудуйте $y = x^2$ для $x = [0, 1, 2, 3, 4, 5]$.

3. Два графіки на одній площині

Накладіть графіки $y = x$ і $y = 2x$ на одні координатні осі. Додайте легенду.

4. Графік температури за тиждень

Маєте дані: дні = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Нд'],

температура = [15, 18, 20, 17, 22, 23, 19].

Побудуйте лінійний графік з підписами осей і заголовком.

5. Налаштування стилю графіка

Побудуйте $y = x^3$ з червоною пунктирною лінією та маркерами у вигляді кіл.

Стовпчасті діаграми (завдання 6–10)

6. Проста стовпчаста діаграма

Маєте: категорії = ['A', 'B', 'C'], значення = [10, 25, 15].

Побудуйте вертикальну стовпчасту діаграму.

7. Горизонтальна стовпчаста діаграма

Ті самі дані, що в завданні 6, але горизонтальна.

8. Діаграма кількості учнів у класах

класи = ['6A', '6B', '6V'], учні = [28, 30, 26].

Додайте підписи на стовпцях (значення).

9. Дві групи на одній діаграмі

Побудуйте стовпчасту діаграму з двома серіями:

чоловіки = [20, 25, 30], жінки = [22, 28, 27] для трьох категорій.

10. Кольори стовпців

Задайте кожному стовпцю свій колір (наприклад, ['red', 'green', 'blue']).

Кругові діаграми (завдання 11–14)

11. Проста кругова діаграма

Маєте: частки = [30, 25, 20, 25], мітки = ['A', 'B', 'C', 'D'].

Побудуйте кругову діаграму.

12. Виділення сектора

У попередній діаграмі виділіть перший сектор («відсуньте» його).

13. Кругова діаграма з відсотками

Додайте на діаграму відсотки (автоматично через `autopct='%1.1f%%'`).

14. Кольори секторів

Задайте власну палітру кольорів для секторів.

Налаштування вигляду (завдання 15–20)

15. Додайте заголовок і підписи осей

Для будь-якого графіка додайте:

`plt.title("Мій графік"), plt.xlabel("X"), plt.ylabel("Y").`

16. Зміна розміру графіка

Зробіть графік ширшим (наприклад, розмір 10x6 дюймів).

17. Сітка на графіку

Увімкніть сітку (`plt.grid(True)`).

18. Збереження графіка у файл

Побудуйте будь-який графік і збережіть його як my_plot.png.

19. Легенда з власними назвами

При побудові двох ліній додайте легенду з назвами "Перший" та "Другий".

20. Графік без рамки та з фоном

Зробіть білий фон, приберіть верхню та праву рамку (spines), залиште лише осі X і Y.

Приклад шаблону для завдань:

```
import matplotlib.pyplot as plt
```

```
x = [0, 1, 2, 3, 4, 5]
```

```
y = [i**2 for i in x]
```

```
plt.plot(x, y, color='red', linestyle='--', marker='o')
```

```
plt.title("Квадрати чисел")
```

```
plt.xlabel("x")
```

```
plt.ylabel("x2")
```

```
plt.grid(True)
```

```
plt.show()
```

Робота з файлами

Завдання на роботу з файлами в Python охоплюють основні операції:

- створення,
- запис,
- читання,
- додавання даних,
- обробка тексту,
- робота з рядками та числами.

Усі завдання використовують вбудовані функції Python (open(), read(), write(), readlines() тощо) і не вимагають додаткових модулів.

Дозволено: with open(...), for line in file, strip(), split(), int(), str().

Не використовувати: pathlib, csv, json, pandas.

Запис у файл (завдання 1–6)

1. Створити файл із привітанням

Створіть файл hello.txt і запишіть у нього рядок: "Привіт, світе!".

2. Записати 5 рядків із числами

Запишіть у файл numbers.txt числа від 1 до 5, кожне в окремому рядку.

3. Запис введеного користувачем тексту

Запитайте у користувача рядок і запишіть його у файл `user_input.txt`.

4. Запис таблиці множення на 7

Запишіть у файл `table7.txt` 10 рядків: $7 * 1 = 7$, $7 * 2 = 14$, ..., $7 * 10 = 70$.

5. Запис квадратів чисел

Запишіть у файл `squares.txt` квадрати чисел від 1 до 10 (кожен у новому рядку).

6. Запис імен у файл

Запитайте 3 імені від користувача і запишіть їх у файл `names.txt`, по одному в рядку.

Читання з файлу (завдання 7–12)

7. Прочитати і вивести весь файл

Прочитайте файл `hello.txt` і виведіть його вміст у консоль.

8. Порахувати кількість рядків у файлі

Прочитайте `numbers.txt` і виведіть, скільки в ньому рядків.

9. Прочитати і вивести кожен рядок окремо

Прочитайте `names.txt` і виведіть кожне ім'я з префіксом "Ім'я: ".

10. Знайти суму чисел у файлі

У файлі `numbers.txt` — числа по одному в рядку. Обчисліть і виведіть їхню суму.

11. Перевірити наявність слова у файлі

Прочитайте `user_input.txt`. Якщо у ньому є слово "Python", виведіть "Знайдено!".

12. Вивести довжину кожного рядка

Для кожного рядка у файлі `names.txt` виведіть його довжину (без символу нового рядка).

Додавання до файлу (завдання 13–15)

13. Дописати нове ім'я у файл

Допишіть у кінець файлу `names.txt` ще одне ім'я (наприклад, "Марія"), з нового рядка.

14. Вести журнал(логування) подій

Створіть файл `log.txt`. Кожного разу при запуску програми дописуйте рядок: "Подія: [поточна дата/час]" (можна просто "Подія N").

15. Дописати кілька чисел

Допишіть у файл `numbers.txt` числа від 6 до 10, кожне з нового рядка.

Комбіновані операції (завдання 16–20)

16. Копіювання файлу

Прочитайте вміст `hello.txt` і запишіть його у новий файл `copy_hello.txt`.

17. Створити файл із парних чисел

Прочитайте `numbers.txt`, відфільтруйте парні числа і запишіть їх у файл `even.txt`.

18. Запис у зворотному порядку

Прочитайте `names.txt`, запишіть його рядки у зворотному порядку у файл `reversed_names.txt`.

19. Об'єднати два файли

Маєте `file1.txt` і `file2.txt`. Створіть `merged.txt`, що містить спочатку вміст першого, потім — другого.

20. Підрахунок слів у файлі

Прочитайте файл `user_input.txt`. Підрахуйте загальну кількість слів (вважайте, що слова розділені пропусками).

Поради:

Завжди використовуйте `with open(...) as f:`, щоб файл автоматично закривався.

Для читання чисел: `int(line.strip())`.

Для роботи з рядками: `line.strip()` видаляє `\n` і пропуски на кінцях.

Якщо файл не існує при читанні — виникне помилка. Для тренування створюйте файли самі.

Зміст

Прості завдання	1
Використовуємо змінні	1
Літерали та змінні, з використанням <code>f</code> -рядків	2
Використання вказівки <code>input()</code> (тільки рядкові змінні)	3
Використання вказівки <code>input()</code> (з <code>int</code> та <code>float</code>)	5
Завдання на обчислення	6
Більш складніші обчислення	7
Виконання вказівок за умовою з використанням <code>if</code> (без <code>else</code>)	9
Виконання за умовою з використанням <code>if-else</code> (без <code>elif</code>)	10
Повна умовна конструкція <code>if-elif-else</code>	12
Цикли з передумовою	15
Цикл <code>while</code> з вказівкою <code>break</code>	16
Цикли з відомою кількістю повторень	18
Різні параметри циклів з відомою кількістю повторень	20
Завдання 1–7: <code>range(stop)</code> (1 параметр)	20
Завдання 8–13: <code>range(start, stop)</code> (2 параметри)	20
Завдання 14–20: <code>range(start, stop, step)</code> (3 параметри)	21
Примітки:	21
Вказівки <code>break</code> і <code>continue</code> в циклах	21
Завдання 1–10: з <code>continue</code> (без списків)	21
Завдання 11–20: з <code>break</code> (без списків)	22
Особливості:	23
Вказівки <code>break</code> і <code>continue</code> в циклах(зі списками)	23
Завдання 1–10: з використанням <code>continue</code>	23
Завдання 11–20: з використанням <code>break</code>	24
Примітки:	25

Вкладені цикли	25	
Геометричні фігури та малюнки (завдання 1–8)	25	
Числові завдання (завдання 9–14)	26	
Робота з рядками (завдання 15–20)	26	
Важливо:	27	
Випадкові числа	27	
Генерація та прості умови (завдання 1–8)	27	
Цикли + випадкові числа (завдання 9–15)	28	
Ігри та моделювання (завдання 16–20)	28	
Підказки:	29	
Графіка з Turtle	29	
Основні фігури (завдання 1–8)	29	
Кольори та стилі (завдання 9–12)	30	
Цикли та повторення (завдання 13–16)	30	
Керування положенням і орієнтацією (завдання 17–20)	30	
Поради:	31	
Графіка у SimpleGraphics	31	
Основні функції SimpleGraphics (пам'ятка):	31	
1. Червоний квадрат	31	
2. Синє коло з жовтою заливкою	32	
3. Прямокутник з написом	32	
4. Три кольорові еліпси	32	
5. Смайлик	32	
6. Діаграма	32	
7. Декартова система координат	32	
8. Трикутник за трьома точками	32	
9. Червоний хрест	32	
10. Мішень	32	
11. Дві фігури з різними кольорами контуру	32	
12. Текст у різних кольорах	32	
13. Прапор України	33	
14. Дорога з розміткою	33	
15. Сніжинка з ліній	33	
16. Шахівниця (4×4)	33	
17. Сонце за горизонтом	33	
18. Графік функції $y = x$	33	
19. Кнопка "Старт"	33	
20. Дві фігури: одна поверх іншої	33	
Використання PyGameZero	33	
Основи: Actor, фон, малювання (завдання 1–7)	34	
🕹️ Управління клавішами (завдання 8–13)	34	
Прості ігрові механіки (завдання 14–20)	35	
Робота з рядковими величинами	35	
Основні операції (завдання 1–7)	36	
Пошук і заміна (завдання 8–13)	36	
✂️ Зрізи та форматування (завдання 14–20)	36	
Підказки для реалізації:	37	
Створення, заповнення та виведення числових списків	37	

Частина 1: Створення та заповнення списків (завдання 1–10)	37	
Частина 2: Виведення списків (завдання 11–20)	38	
Поради:	39	
Пошук елемента у списку за певним критерієм		39
Числові списки (завдання 1–12)	39	
Рядкові списки (завдання 13–20)	40	
Рекомендації:	41	
Обчислення підсумкових значень елементів списку		41
Сума та кількість (завдання 1–7)	41	
Добуток та інші операції (завдання 8–12)	41	
Середнє, максимум, мінімум (завдання 13–20)	42	
Поради:	42	
Сортування списків		43
Сортування за зростанням (завдання 1–7)	43	
Сортування за спаданням (завдання 8–12)	43	
Складніші критерії (завдання 13–20)	44	
Поради:	44	
Випадкові величини та списки		45
Заповнення списків (завдання 1–6)	45	
Аналіз списку (завдання 7–13)	45	
Перемішування, вибір, зміна (завдання 14–20)	46	
Корисні підказки:	46	
Створення програм з графічним інтерфейсом		47
Основи: вікно та прості віджети (завдання 1–6)	47	
Взаємодія: обробка подій (завдання 7–12)	47	
Компонування та компоненти (завдання 13–17)	48	
Міні-проєкти (завдання 18–20)	48	
Поради:	48	
Побудова діаграм і графіків		49
Лінійні графіки (завдання 1–5)	49	
Стовпчасті діаграми (завдання 6–10)	49	
Кругові діаграми (завдання 11–14)	50	
Налаштування вигляду (завдання 15–20)	50	
Приклад шаблону для завдань:	50	
Робота з файлами		51
Запис у файл (завдання 1–6)	51	
Читання з файлу (завдання 7–12)	51	
Додавання до файлу (завдання 13–15)	52	
Комбіновані операції (завдання 16–20)	52	
Поради:	52	

