

**Алгоритм** — це запис скінченої послідовності вказівок, виконання яких призводить до розв'язання певної задачі.

**Вказівка** (алгоритму) — це спонукальне речення, що вказує, яку дію має виконати виконавець алгоритму.

**Виконавець** (алгоритму) — це жива істота (людина або тварина) або автоматичний пристрій (робот, електронна обчислювальна машина тощо), спроможна діяти відповідно з алгоритмом.

**Система вказівок виконавця** — це множина (сукупність) всіх вказівок, які може виконувати даний виконавець.

**Середовище виконання алгоритму** — об'єкти, з якими працює виконавець у процесі виконання алгоритму.

**Властивості алгоритму:** дискретність, визначеність, виконуваність, скінченність, результативність, масовість, ефективність.

Дискретність (латинською *discretus* — розділений, розривний) алгоритму означає, що виконання алгоритму зводиться до виконання окремих дій (кроків) у певній послідовності. Причому, кожному вказівку алгоритму виконують за скінченний проміжок часу.

Визначеність (однозначність) означає, що алгоритм однозначно визначає порядок дій виконавця, результат цих дій і не потребує додаткового тлумачення..

Виконуваність означає, що алгоритм, призначений для певного виконавця, може містити лише вказівки, які входять до системи вказівок цього виконавця.

Скінченність означає, що виконання алгоритму закінчиться після скінченної (можливо, досить великої) кількості кроків і за скінченний час для довільних вхідних даних.

Результативність алгоритму означає, що після закінчення виконання алгоритму обов'язково:

або одержують розв'язок-результат відповідно до поставленої мети;

або встановлюють відсутність таких розв'язків (множина розв'язків порожня);

або встановлюють неможливість розв'язання задачі.

Масовість алгоритму означає, що алгоритм можна застосувати до цілого класу однотипних задач, для яких спільними є умова та хід розв'язування та які відрізняються лише початковими (вхідними) даними. Наприклад, алгоритмом дій, складеним для одного касира, можуть успішно скористатися всі касири супермаркету. А програмою пошуку коду і підрахунку суми вартостей товарів, придбаних покупцем, — усі комп'ютери супермаркету.

Ефективність алгоритму описує час виконання і об'єм ресурсів, необхідних для виконання алгоритму: чим менше часу (часова ефективність) і ресурсів (просторова ефективність), тим ефективність вища.

Потреба запровадження табличних величин впливає з необхідності запам'ятовувати й однотипно (за одним алгоритмом) опрацьовувати великий набір можливо однотипних даних. Наприклад, при знаходженні середнього балу кожного учня класу з інформатики за чверть потрібно знаходити суму великої кількості оцінок. Як зберігати всі ці оцінки? Зарезервувати для цього 40 чи більше змінних? Це дуже незручно.

У мові Python відповідні зручні структури даних називають послідовностями. Це рядки (символів), списки, масиви, значення функції `range` і ще деякі інші об'єкти.

Списку (у мові програмування) відповідає (у математиці) послідовність, члени якої занумеровано невід'ємними цілими числами, починаючи з 0, з нефіксованими довжиною та типом елементів.

## ВВЕДЕННЯ І ВИВІД ЕЛЕМЕНТІВ СПИСКУ

Для введення елементів списку в процесі виконання програми використовують функцію **input()** і метод **split()**.  
**list=input().split()**

```
1 a= input().split()
2 print (a)
```

1 2 3 4 5  
['1', '2', '3', '4', '5']

```
1 a= [int(i) for i in input().split()]
2 print (a)
```

1 2 3 4 5  
[1, 2, 3, 4, 5]

Вивести всі елементи списку а можна за допомогою команди **print(a)**.  
При цьому буде виведено значення елементів списку, поміщені в квадратні дужки, через кому.

```
a = [1, 2, 3]
b = [4, 5]
c = a + b
d = b * 3
print(c)           [1, 2, 3, 4, 5]
print(d)           [4, 5, 4, 5, 4, 5]
print([7, 8] + [9]) [7, 8, 9]
print([0, 1] * 3)  [0, 1, 0, 1, 0, 1]
```

### Вивід елементів списку в стовпець

```
1 # Вивід елементів списку в стовпець
2 a=[5, 2, 7, 4, 3]
3 for i in range(len(a)):
4     print(a[i])
```

```
>>> %Run
5
2
7
4
3
```

### Вивід елементів списку рядком

```
1 # Вивід елементів списку рядком
2 a=[5, 2, 7, 4, 3]
3 for elem in a:
4     print(elem, end=' ')
```

```
>>> %Run
5 2 7 4 3
```