



Сергій Матвійчук

# АБЕТКА ПРОГРАМУВАННЯ на Python

збірник задач вміщує пояснення алгоритмів та  
коди програм до 200 окремих задач  
з сайту **e-olymp.com**

2020



.....	1
<b>ЛІНІЙНІ ПРОГРАМИ</b> .....	<b>4</b>
<b>ЛІНІЙНІ ПРОГРАМИ. ПОЧАТОК</b> .....	<b>6</b>
<b>ЛІНІЙНІ ПРОГРАМИ. ОБЧИСЛЕННЯ</b> .....	<b>8</b>
<b>ОБЧИСЛЕННЯ ВИРАЗІВ І ФУНКЦІЙ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ДІЛЕННЯ ЦІЛИХ ЧИСЕЛ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ДІЛЕННЯ ЦІЛИХ ЧИСЕЛ 2</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>РОЗГАЛУЖЕННЯ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З РОЗГАЛУЖЕННЯМ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З РОЗГАЛУЖЕННЯМ 2</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З РОЗГАЛУЖЕННЯМ 3</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЦИКЛ WHILE З ПЕРЕДУМОВОЮ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З ЦИКЛОМ WHILE</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З ЦИКЛОМ WHILE 2</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З ЦИКЛОМ WHILE 3</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЦИКЛ FOR</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З ЦИКЛОМ FOR</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ З ЦИКЛОМ FOR 2</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>МАСИВИ АБО СПИСКИ</b> .....	<b>12</b>
<b>ЗАДАЧІ НА МАСИВАХ</b> .....	<b>15</b>
<b>ЗАДАЧІ НА МАСИВАХ 2</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ОБРОБКА РЯДКІВ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ ПРО РЯДКИ</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
<b>ЗАДАЧІ ПРО РЯДКИ 2</b> .....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.

Близько року тому на сайті **e-olymp.com** з'явилась добірка завдань початкового рівня під загальною назвою "Абетка програмування" це номери 8800..9000. Коли кількість задач на сайті сягає декількох тисяч, то знайти потрібний набір, тим більш початкового рівня досить проблематично. Задачі з "Абетки" розміщені методологічно і відповідно до шкільної програми, щоб допомогти тим, хто робить перші кроки в "школі молодого бійця". Тисячі зарахованих і незарахованих спроб на цих завданнях від користувачів сайту свідчать про популярність і корисність завдань.

Завдання "Абетки" створювались без попередньої підготовки, тобто умови задач, тести, коди для перевірки писалися прямо на сайт. Може дещо поспішно, але так вже вийшло. Тому постала необхідність в посібнику, де можна прочитати умови завдань в хронологічному порядку, мати необхідний довідковий матеріал та для прикладу розв'язання деяких з них.

Найближчим часом всі завдання "Абетки" були перенесені у Word і до Вашої уваги посібник або збірник задач "Абетка програмування" тільки з "вогню-полум'я". Записані всі двісті задач в одному з двох варіантів:

- номер з сайту, назва, умова та приклад тесту;
- номер з сайту, назва, умова, приклад тесту та коротке пояснення і код програми.

Задачі розбиті (майже завжди) на групи по дванадцять завдань, групи відповідно об'єднані в розділи "Лінійні програми", "Розгалуження" "Цикл **while**", "Цикл **for**", "Масиви", "Рядки". Кожен розділ має коротенький теоретичний вступ та довідковий матеріал по **Python** – скопійовано з книги «Практикум програмування Python / C++ на e-olymp.com». Нічого зайвого, мінімальна кількість символів і аркушів, їх до речі 75. Розраховано, щоб вчитель мав змогу роздрукувати посібник в необхідній кількості для учасників навчального процесу.

Користуватися посібником можна в декількох варіантах:

- просто розв'язувати завдання, не використовуючи сайт **e-olymp**;
- розв'язувати завдання, використовуючи для перевірки програм сайт;
- отримати статус керівника груп на сайті **e-olymp**, проводити підсумкові і тренувальні роботи автоматизованими засобами, що особливо актуально в наш нелегкий час дистанційного навчання і карантину.

Бажаю успіхів. Надіюся, що ця книга буде Вам корисною.

# ЛІНІЙНІ ПРОГРАМИ

Розпочнемо з вправ, які традиційно називають лінійними програмами – тобто не потребують особливих методів і прийомів програмування, а мають три чітко окреслених блоки, які послідовно виконуються в програмі – це ввід даних, обчислення та вивід результатів. Зазвичай, в таких програмах проблематичною є друга, тобто математична частина. Зрозуміло, що кожного разу будуть різні формули, їх можна придумати використовуючи свій математичний досвід або глянути в Інтернеті. Потрібно не забувати, що Ви розробник програми і невірно вибрана математична модель буде давати неправильні результати або не працювати взагалі. Короткі коментарі по інструкціях в лінійних програмах, тільки тези:

- для друку чисел, тексту і значень змінних в мові Python використовується оператор **print()**;
- щоб вивести текст, наприклад **Hello!**, його записують в круглих дужках і в лапках після **print**, ось так **print("Hello!")**;
- щоб вивести числове значення, в круглих дужках після **print()** записується константа, ім'я змінної або арифметичний вираз;
- щоб вивести декілька об'єктів, їх записують аргументами в інструкції **print()** через кому;
- імена змінних в Python відповідають загальноприйнятим правилам в мовах програмування, тобто позначаються буквами і цифрами (перша буква), але великі і маленькі букви розрізняються;
- типи змінних попередньо прописувати не потрібно;
- під час виконання програми одна та ж сама змінна може приймати значення різних типів в залежності від інструкцій;
- для перетворення типів значень змінних використовуються функції **int()**, **float()**, **bool()**, **str()**;
- ввід даних в Python виконує оператор **input()**, причому введене значення рахується текстовим рядком;
- для введення чисел використовуються інструкції **int(input())** для цілих або **float(input())** для дійсних;
- вирази в командах присвоювання і друку використовують загальноприйнятий порядок дій, арифметичні операції і математичні функції;
- схема лінійного алгоритму - перш за все потрібно ввести дані, потім обчислити і тільки на кінець вивести результати.

Якщо поки не все зрозуміло, не біда. Можете сміливо приступати до розгляду конкретних задач, а тези перечитаєте ще раз пізніше.

# Довідник Python

Основні арифметичні операції	
додавання	<b>+</b>
віднімання	<b>-</b>
множення	<b>*</b>
ділення	<b>/</b>
піднесення до степені	<b>**</b>
ділення націло	<b>//</b>
остача від ділення націло	<b>%</b>
Основні арифметичні функції	
модуль аргументу <b>x</b>	<b>abs(x)</b>
перетворення рядка <b>x</b> до цілого	<b>int(x)</b>
перетворення рядка <b>x</b> в системі числення <b>n</b> до цілого	<b>int(x, n)</b>
перетворення аргументу до дійсного	<b>float(x)</b>
округлення аргументу до <b>n</b> знаків після коми	<b>round(x, n)</b>
перетворення цілого числа в двійковий рядок	<b>bin(x)</b>
перетворення цілого числа в вісімковий рядок	<b>oct(x)</b>
перетворення цілого числа в шістнадцятьковий рядок	<b>hex(x)</b>
повертає частку і остачу від ділення <b>a</b> на <b>b</b>	<b>divmod(a, b)</b>
повертає <b>a</b> в степені <b>b</b>	<b>pow(a, b)</b>
Основні команди в лінійних програмах	
ввести текстовий рядок з консолі в <b>x</b>	<b>x = input()</b>
ввести ціле число в <b>X</b>	<b>x = int(input())</b>
ввести дійсне число в <b>X</b>	<b>x = float(input())</b>
ввести декілька цілих чисел в <b>X, Y</b>	<b>x, y = map(int, input().split())</b>
ввести декілька дійсних чисел в <b>X, Y</b>	<b>x, y = map(float, input().split())</b>
вивести значення <b>X</b>	<b>print(x)</b>
вивести значення <b>X, Y</b>	<b>print(x, y)</b>

# ЛІНІЙНІ ПРОГРАМИ. ПОЧАТОК.

## 8800 Hello, Python!

Виведіть повідомлення **Hello,Python!**

Hello, Python!
----------------

Текст виводить команда <b>print</b> , потрібно уважно ставити дужки і лапки.
--

<pre>print("Hello, Python!")</pre>
------------------------------------

## 8801 Наступне число

Прочитати ціле число та вивести наступне за ним число.

10	11
----	----

Програма читає з консолі ціле число <b>n</b> і виводить на екран значення <b>n+1</b> .
--

<pre>n=int(input()) print(n+1)</pre>
--

## 8802 Попереднє число

Програма повинна прочитати з консолі ціле число та вивести попереднє до нього число.

10	9
----	---

## 8803 Хто крайній?

Оленка є передостанньою у списку учнів свого класу. Знайти кількість учнів, якщо відомо номер Оленки.

20	21
----	----

## 8804 Сума двох цілих

На вході програми маємо два цілих числа, кожне в окремому рядку. На вихід потрібно подати суму заданих чисел.

12	21
9	

Програма читає з консолі два цілих числа <b>n</b> та <b>m</b> , кожне з окремого рядка та виводить на екран їх суму.
--

<pre>n=int(input()) m=int(input()) print(n+m)</pre>
---

## 8805 Сума двох цілих 2

На вході програми маємо два цілих числа, записані в одному рядку через пропуск. На вихід потрібно подати суму заданих чисел.

12 9	21
------	----

Аналогічно попередньому завданню. Варто звернути увагу на особливості введення двох цілих чисел, записаних в одному рядку через проміжок.

```
a,b=map(int,input().split())  
print(a+b)
```

## 8806 Кількість учнів

В класі навчається **a** хлопчиків і **b** дівчат. Скільки всього учнів в класі?

12 9	21
------	----

## 8807 Протилежне число

Програма повинна прочитати з консолі ціле число та вивести протилежне до нього число.

10	-10
----	-----

## 8808 Різниця двох цілих

На вході програми маємо два цілих числа, кожне в окремому рядку. На вихід потрібно подати різницю між першим і другим числом.

21 9	12
------	----

## 8809 Марафон

Змагання з бігу розпочали **a** учасників, але **b** з них зійшли з дистанції. Скільки бігунів фінішували?

12 9	3
------	---

## 8810 Шкільний концерт

На шкільному концерті **a** учнів співали, **b** - танцювали, а **c** – співали й танцювали. Скільки було учасників всього?

13 9 5	17
--------	----

Від суми двох перших чисел потрібно відняти третє, бо значення **c** врахували двічі.

```
a,b,c=map(int,input()  
) .split()  
print(a+b-c)
```

### 8811 Добуток двох цілих

На вході програми маємо два цілих числа, записані в одному рядку через пропуск. На вихід потрібно подати добуток заданих чисел.

3 7	21
-----	----

## ЛІНІЙНІ ПРОГРАМИ. ОБЧИСЛЕННЯ.

### 8812 Периметр і площа

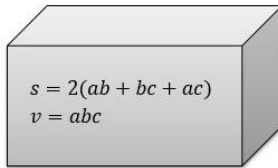
Знайти периметр і площу прямокутника, якщо відомі довжини його сторін.

$$p = 2(a + b)$$
$$s = ab$$

3 7	20 21
-----	-------

### 8813 Площа поверхні та об'єм

Знайти площу поверхні та об'єм прямокутного паралелепіпеда, якщо відомі його виміри.


$$s = 2(ab + bc + ac)$$
$$v = abc$$

2 3 4	52 24
-------	-------

Алгоритм очевидний: – вводимо $a, b, c$ – обчислюємо $s, v$ – виводимо $s, v$	<pre>a,b,c=map(int,input().split()) s=2*(a*b+b*c+a*c) v=a*b*c print(s,v)</pre>
--	--

### 8814 Периметр і площа 2

Знайти периметр і площу квадрата, якщо відома довжина його сторони  $a$ .

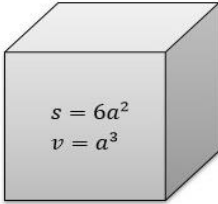
$$p = 4a$$
$$s = a^2$$

3	12 9
---	------



### 8815 Площа поверхні та об'єм 2

Знайти площу поверхні та об'єм куба, якщо відома довжина його ребра.



3	54 27
---	-------

### 8816 Степінь числа

На вході програми маємо два натуральних числа **a** та **n**, кожне в окремому рядку. На вихід потрібно подати **a<sup>n</sup>**.

2 10	1024
------	------

### 8817 Кількість n-значних чисел

На вході програми маємо натуральне число **n**. Вивести кількість **n**-значних натуральних чисел.

1	9
---	---

Кількість <b>n</b> -значних натуральних чисел можна знайти за формулою $9 \cdot 10^{n-1}$ . У найстаршому розряді може будь-яка цифра крім нуля, тобто дев'ять різних і всі десять цифр в інших розрядах.	<pre>n=int(input()) print(9*10**(n-1))</pre>
---	--

### 8818 Кількість непарних n-значних чисел

На вході програми маємо натуральне число **n**. Вивести кількість непарних **n**-значних натуральних чисел.

1	5
---	---

### 8819 Кількість парних n-значних чисел

На вході програми маємо натуральне число **n**. Вивести кількість парних **n**-значних натуральних чисел.

1	4
---	---

Очевидно, що парних буде половина від загальної кількості чисел.	<pre>n=int(input())-1 print(9*10**n//2)</pre>
--	---

## 8820 Кількість n-значних чисел 2

На вході програми маємо натуральне число **n**. У відповідь потрібно вивести кількість **n**-значних натуральних чисел, у записі яких використано тільки непарні цифри.

1	5
---	---

## 8821 Кількість n-значних чисел 3

На вході програми маємо натуральне число **n**. У відповідь потрібно вивести кількість **n**-значних натуральних чисел, у записі яких використано тільки парні цифри.

1	4
---	---

Найстарший розряд кожного з таких чисел можна заповнити однією з чотирьох цифр - 2,4,6,8 (цифра 0 не підходить). Для наступних розрядів підходять п'ять цифр. Отже, актуальна формула $4*5^{n-1}$ .	<pre>n=int(input()) print(4*5**(n-1))</pre>
---	---

## 8822 Кількість n-значних чисел 4

На вході програми маємо натуральне **n**. У відповідь потрібно вивести кількість **n**-значних натуральних чисел, в записі яких немає жодної цифри **6**.

1	4
---	---

Якщо цифру <b>6</b> не використовувати, то очевидно, що найстарший розряд вибираємо з <b>8</b> цифр (всі крім <b>0</b> і <b>6</b> ). Наступні розряди мають по <b>9</b> варіантів. Отже, дійсна формула $8*9^{n-1}$ .
---

## 8823 Кількість n-значних чисел 5

На вході програми маємо натуральне **n**. У відповідь потрібно вивести кількість **n**-значних натуральних чисел, в записі яких є хоча б одна цифра **7**.

1	1
2	18

Очевидно, має місце алгоритм :

- кількість всіх **n**-значних натуральних чисел дорівнює  $9*10^{n-1}$ ;
- кількість **n**-значних чисел, що не містять **7** дорівнює  $8*9^{n-1}$ ;
- якщо від першого результату відняти другий, то отримаємо відповідь до задачі (до речі, для будь-якої іншої ненульової цифри результат буде аналогічний).

## 8824 Знайти число

Знайти ціле число  $n$ , якщо відомі попереднє та наступне до нього числа, записані у довільному порядку через пропуск.

6 8	7
-----	---

Будь-яке ціле число дорівнює середньому арифметичному своїх сусідів. І ще, виконуючи переніс команди в наступний рядок, достатньо розірвати пару відповідних дужок.	<pre>a,b=map(int, input().split( )) print((a+b)//2)</pre>
---	---

# МАСИВИ АБО СПИСКИ

Масив – це дещо схоже на багатоквартирний будинок, аркуш табличного процесора Excel або ж оперативну пам'ять комп'ютера, коли по заданому адресу завжди знаходимо деякий об'єкт, подібний до всіх інших. Причому всі елементи масиву, зазвичай (але необов'язково), мають однаковий тип і позначаються одним ім'ям, різні тільки індекси, які починаються з **0** і йдуть під-ряд. Однакові значення можуть входити в масив багаторазово.

Перегляд елементів масиву найчастіше здійснюється командою циклу з параметром **for** двома способами:

- по елементах **for q in a**, де **q** пробігає послідовно всі елементи масиву **a**;
- по індексах **for i in range(n)**, де **a[i]** (**i=0..n-1**) пробігає послідовно всі **n** елементів масиву **a**;

Введення масиву, елементи якого записані в стрічку через проміжок можна виконати такими інструкціями Python:

- **a = list(input().split())** – введення текстового масиву **a**;
- **a = [v for v in input().split()]** – аналогічно попередньому;
- **a = list(map(int, input().split()))**– введення масиву цілих чисел **a**;
- **a = [int(i) for i in input().split()]** – аналогічно попередньому.

Виведення всього масиву виконується командою **print(a)**, для поелементного виведення в стрічку через проміжок або в стовпчик команда **print()** викликається в циклі **for** для кожного елемента масиву.

Для кращого розуміння пропонується програма, яка дозволяє ввести з консолі масив цілих чисел. А далі виводить масив, сортує елементи у зростанні і виводить їх в стовпчик. І ще, розвертає елементи масиву в протилежному напрямку і виводить елементи масиву в стрічку через проміжок. Кількість елементів масиву в такій версії програми вводити не потрібно.

```
a=list(map(int,input().split()))
print(a)
a.sort()
for q in a : print(q)
a.reverse()
for q in a : print(q,end=' ')
```

# Довідник Python

Методи і функції пов'язані з масивами	
Створення списку об'єктів (масиву).	<code>x=list()</code>
Визначення довжини списку (масиву).	<code>len(x)</code>
Додає елемент <code>z</code> в кінець списку <code>x</code> .	<code>x.append(z)</code>
Видаляє перший елемент у списку <code>x</code> , який має значення <code>z</code> .	<code>x.remove(z)</code>
Вставляє на <code>i</code> -ий елемент значення <code>z</code> .	<code>x.insert(i,z)</code>
Видаляє <code>i</code> -ий елемент і повертає його. Якщо індекс не вказано, видаляється останній.	<code>x.pop(i)</code>
Повертає положення першого елемента, значення якого <code>z</code> .	<code>x.index(z)</code>
Повертає кількість елементів, що дорівнюють <code>z</code> .	<code>x.count(z)</code>
Сортує список в неспадному порядку.	<code>x.sort()</code>
Розвертає елементи списку.	<code>x.reverse()</code>
Очищає список, видаляє всі його елементи.	<code>x.clean()</code>
Перетворення списку у множину.	<code>set(x)</code>

Задач, що використовують масиви в своїх розв'язках на порядок більше ніж завдань, з яких можна засвоїти початкові поняття: введення, виведення елементів, підрахунок суми, добутку, пошук, зміна порядку, сортування елементів масиву. Адже, громіздка програма для розв'язання складної задачі повинна складатись з максимально лаконічно і точно описаних кубиків коду, тоді можна говорити про оптимальність і ефективність алгоритму. Не вдається створити гарний будинок з бракованих матеріалів, він швидко зруйнується від холоду, дощу і вітру.

На сайті знаходимо багато задач для покращення навиків роботи з одновимірними, а також двовимірними масивами. Можна запропонувати до уваги список задач з розділу «Просто масиви» з книги «Практикум програмування Python / C++ на e-olymp.com». Корисні завдання, скоріше розраховані на використання готових процедур і функцій **Python** для обробки списків. Навіть,

якщо розглянути тільки табличку вище, можна переконатись, що **Python** надає масу засобів для ефектної роботи в такому напрямку.

- 7829 Сума елементів масиву
- 7830 Найбільший елемент масиву
- 7832 Кількість максимальних
- 7831 Сума без максимального
- 7841 Непарні елементи
- 7842 Парні індекси
- 7843 Парні індекси
- 7844 Сусіди одного знаку
- 7845 Більші своїх сусідів
- 7846 Найбільший елемент
- 7847 Кількість різних елементів
- 7848 Переставити сусідні
- 7849 Обміняти max і min
- 7850 Обміняти max і min
- 7833 Більші за середнє арифметичне
- 7834 Два найбільших
- 5337 Різні-різні
- 2440 N-те найбільше значення
- 5721 Пошук елемента
- 3935 Реверс
- 2099 Два масиви

Не має границь фантазія і творчість у справжніх програмістів, як і немає такого програмного коду, якого не можливо було б покращити і оптимізувати. А якщо не знайшлося потрібного інструменту у списку стандартних процедурі функцій, отже потрібно вміти дещо написати вручну. Саме про це наступний набір задач в «Абетці програмування», тут і зараз можна познайомитись з доповненням і продовженням розділу «Просто масиви».

# ЗАДАЧІ НА МАСИВАХ

## 8953 Вивести масив

Задано масив з  $n$  цілих чисел. Вивести його елементи в стовпчик, не змінюючи початковий порядок.

4	5
5 0 -7 2	0
	-7
	2

Читаємо значення  $n$  та цілі числа з одного рядка через проміжок, як елементи масиву  $a$ . Виводимо значення елементів циклом **for**, переглядаючи їх по елементах.

```
n=int(input())
a=[int(i) for i in input(
).split()]
for j in a : print(j)
```

## 8954 Вивести масив 2

Програма має прочитати з консолі масив з  $n$  цілих чисел та вивести елементи масиву в одному рядку через проміжок, змінивши початковий порядок на протилежний.

7	-2 3 0 -4 7 4 0
0 4 7 -4 0 3 -2	

Читаємо значення  $n$  та утворюємо масив  $a$  з  $n$  нулів. Вводимо значення елементів масиву – кожне з окремого рядка, переглядаючи їх по індексах. Виводимо масив в один рядок в оберненому порядку.

```
n=int(input())
a=[0]*n
for i in range(n) :
    a[i]=int(input())
for i in range(n-1,-1,-1) :
    print(a[i], end=' ')
```

## 8955 Вивести масив 3

Задано масив з  $n$  цілих чисел. Вивести тільки додатні його елементи, не змінюючи їх початковий порядок або повідомлення **NO**, якщо їх немає.

7	3
-2 5 4 -3 7 -3 0	5 4 7
5	NO
-2 -1 0 -1 -5	

## 8956 Вивести масив 4

Програма має прочитати з консолі масив з  $n$  цілих чисел та вивести тільки від'ємні елементи цього масиву одним рядком через проміжок, змінивши початковий порядок на протилежний або повідомлення **NO**, якщо їх немає.

7 -2 5 4 -3 7 -1 0	3 -1 -3 -2
5 2 1 0 1 5	NO

Стандартний ввід масиву <b>a</b> . Шукаємо кількість від'ємних в змінну <b>k</b> . Виводимо <b>NO</b> , якщо не знайшли або виводимо <b>k</b> і потім всі від'ємні елементи, переглядаючи їх по індексах в оберненому порядку.	<pre>n,k=int(input()),0 a=[int(j) for j in input( ).split()] for j in a :     if j&lt;0 : k+=1 if k==0 :     print('NO') else :     print(k)     for i in range(n-1,-1,-1) :         if a[i]&lt;0 :             print(a[i], end=' ')</pre>
--	--

## 8957 Вивести масив 5

Програма має прочитати з консолі масив з  $n$  цілих чисел та вивести тільки парні елементи цього масиву одним рядком через проміжок, змінивши початковий порядок на протилежний або повідомлення **NO**, якщо їх немає.

7 -2 5 4 -3 7 -1 0	3 0 4 -2
-----------------------	-------------

## 8958 Вивести масив 6

Програма має прочитати з консолі масив з  $n$  цілих чисел та вивести тільки елементи з непарними індексами одним рядком через проміжок, не змінюючи початковий порядок або повідомлення **NO**, якщо їх немає.. Нумерація починається з **0**.

7 -2 5 4 -3 7 -1 0	3 5 -3 -1
-----------------------	--------------



Стандартний ввід масиву **a**. Елементів з непарними індексами буде **k=n//2**. Виводимо **NO**, якщо **k=0** або виводимо **k**, а після всі шукані елементи, переглядаючи їх по індексах від **1** до **n** з кроком **2**.

```
n,k=int(input()),0
a=[int(j) for j in input(
).split()]
k=n//2
if k==0 :
    print('NO')
else :
    print(k)
    for i in range(1,n,2) :
        print(a[i], end=' ')
```

### 8959 Різниця між найбільшим і найменшим

Задано **n** цілих чисел. Вивести різницю між найбільшим і найменшим числом.

7	27
0 -7 -13 14 -2 13 13	

### 8960 Крім найменших і найбільших

З консолі вводиться масив з **N** цілих чисел. Програма повинна знайти суму елементів масиву, не враховуючи всіх його найменших і найбільших елементів.

7	10
6 2 7 1 7 1 2	

Стандартний ввід масиву **a**. Далі скористаємося стандартними функціями **Python** для масивів. Особливий випадок, коли **min=max**.

```
n=int(input())
a=[int(j) for j in input().split()]
l,m=min(a),max(a)
p=a.count(l)
q=a.count(m)
if l==m :
    print(sum(a)-l*p)
else :
    print(sum(a)-l*p-m*q)
```

### 8961 Перший найменший

Задано масив з **n** цілих чисел. Знайдіть найменший елемент масиву, що зустрічається найпершим та поміняйте його з першим елементом у масиві, не змінюючи порядок інших елементів.

7	-5 -3 5 6 -4 7 -5
6 -3 5 -5 -4 7 -5	

## 8962 Крайній найбільший

Задано масив з  $n$  цілих чисел. Знайдіть найбільший елемент масиву, що зустрічається останнім та поміняйте його з крайнім елементом у масиві, не змінюючи порядок інших елементів.

7 6 -3 7 -4 7 4 -5	6 -3 7 -4 -5 4 7
-----------------------	------------------

Після стандартного вводу масиву **a** знаходимо номер крайнього найбільшого елементу масиву. Виконуємо обмін значень у вказаних елементів масиву **a**. Виявляється вивід масиву можна однією командою **print(\*a)** – вивести вміст пам'яті, на яку вказує ім'я **a**.

```
n=int(input())
a=[int(j) for j in input(
).split()]
m=max(a)
for i in range(n) :
    if a[i]==m : k=i
a[k]=a[n-1]; a[n-1]=m
print(*a)
```

## 8963 Найменші вліво

Задано масив з  $n$  цілих чисел. Перемістити всі найменші елементи на початок масиву, не змінюючи порядок інших.

7 6 -3 -7 4 -7 -4 5	-7 -7 6 -3 4 -4 5
------------------------	-------------------

## 8964 Найбільші вправо

Задано масив з  $n$  цілих чисел. Перемістити всі найбільші елементи в кінець масиву, не змінюючи порядок інших.

7 6 -2 7 1 7 -1 2	6 -2 1 -1 2 7 7
----------------------	-----------------