

Як робити домашні завдання

Формат програми

Розв'язком кожної задачі має бути консольна програма, яка читає дані з вказаного в умові файлу, і записує результати у вказаний в умові файл.

1. Не потрібно робити ввід/вивід з клавіатури чи графічний інтерфейс.
2. Імена файлів вводу/виводу можна «хардкодити» в програмі, проте вказуйте відносні, а не абсолютні шляхи. Розраховуйте на те, що вхідні та вихідні файли знаходяться у тій самій папці, що й виконувана програма.

Корисно: Щоб легше було відлагоджувати програму і тестувати її на різних наборах даних, ви можете (на власний розсуд) передбачити додаткову функцію: якщо програмі були передані імена файлів в командному рядку, тоді використовувати їх, інакше — імена за замовчуванням. Приклад:

- `"discnt.exe"` — читатиме дані з `discnt.in` та виводитиме результати в `discnt.out`.
 - `"discnt.exe case1.in case1.out"` — читатиме дані з `case1.in` та виводитиме результати в `case1.out`.
3. Коректність вхідних даних не потрібно перевіряти.
 4. Формат чисел для вхідних/вихідних даних:
 - Розділювачів тисяч немає. Наприклад, 5000, 1000000, 123456789000.
 - Дробовий знак — крапка. Наприклад, 0.333, 12.75, 9.0.
 5. Кодування вхідних та вихідних файлів — UTF-8 (without BOM), якщо інше не вказано в умові.

Як здавати рішення

1. Якщо ви ще не створили репозиторію на GitHub, створіть його. Ім'я може бути довільне, наприклад, `lits-ads-002`.
2. Зверніть увагу на 6-символьний код задачі, вказаний в її описі. Наприклад, DISCNT.
3. Для кожної задачі створюйте окрему папку, наприклад `/lits-ads-002/discnt`. Назва папки обов'язково повинна співпадати з 6-символьною назвою задачі.
4. В кожній папці задачі зберігайте ваш програмний код. Наприклад, `/lits-ads-002/discnt/discnt.py`.
Дозволяється мати вкладені папки (наприклад, `/lits-ads-002/discnt/src/Discnt.java`) та більше, ніж один файл з кодом, проте бажано, щоб головна програма називалася так само, як задача. Скомпільованих бінарних файлів зберігати не потрібно (використовуйте файл `.gitignore`: детальніше — [документація](#), [зразки на GitHub](#))
5. Ваші рішення будуть автоматично тестуватися кожні 3 години. Слідкуйте за сторінкою <http://lits-ads-002.s3-website.eu-central-1.amazonaws.com/>, щоб дізнатися про результати.

(Зразок задачі)



Код задачі: MINMAX

Знайдіть у масиві найменше та найбільше число.

Вхідні дані

Вхідний файл `minmax.in` складається з одного рядка, який містить перелік цілих чисел від 0 до 10^{15} включно, розділених пробілом — елементи масиву. Загальна кількість елементів може бути від 1 до 1 000 000.

Вихідні дані

Вихідний файл `minmax.out` повинен містити два числа, розділені пробілом — значення мінімального та максимального елементів.

Приклад 1

minmax.in

50 20 30 47 100

minmax.out

20 100

Приклад 2

minmax.in

1

minmax.out

1 1

Зразок розв'язку (Python 2)

```
import sys

def main():
    input_filename = "minmax.in" if len(sys.argv) == 1 else sys.argv[1]
    output_filename = "minmax.out" if len(sys.argv) == 1 else sys.argv[2]

    array = read_input(input_filename)
    minimum, maximum = solve(array)
    write_output(output_filename, minimum, maximum)

def read_input(filename):
    with open(filename, "r") as input_file:
        array_str = input_file.readline()
        array = [int(item) for item in array_str.split()]
        return array

def solve(array):
    minimum = min(array)
    maximum = max(array)
    return minimum, maximum

def write_output(filename, minimum, maximum):
    with open(filename, "w") as output_file:
        output_file.write("{minimum} {maximum}".format(**locals()))

if __name__ == "__main__":
    main()
```