

## Задача А. «Операції над множинами»

Вхідні дані: Або клавіатура, або input.txt      Обмеження часу: 1 сек  
Результати: Або екран, або output.txt      Обмеження пам'яті: 64 мегабайти

Як відомо, множина — це сукупність елементів, що розглядається (вся сукупність) як єдине ціле. Зазвичай вважають, що множина не може містити один і той самий елемент кілька разів (лише або містить, або ні). Взагалі кажучи, елементи можуть бути різної природи, але у цій задачі елементами будуть лише цілі невід'ємні числа. Для множин відомі деякі стандартні операції. Ми розглянемо лише три найстандартніші з них:

**Об'єднання** (математичне позначення " $\cup$ ", у нас позначається UNION) двох множин  $A$  та  $B$  — це множина усіх елементів, що належать хоча б одній з множин  $A$  або  $B$ . Якщо елемент належить обом множинам, він все одно враховується один раз.

**Перетин** (математичне позначення " $\cap$ ", у нас позначається INTERSECTION) двох множин  $A$  та  $B$  — це множина усіх елементів, що належать обом множинам  $A$  та  $B$  одночасно.

**Різниця** (математичне позначення " $\setminus$ " (зворотня коса риска), у нас позначається DIFFERENCE) двох множин  $A$  та  $B$  — це множина усіх елементів, що належать множині  $A$ , але не належать  $B$ . Ця операція, єдина з трьох згаданих, несиметрична (не комутативна).

Напишіть програму, яка виконуватиме ці операції над множинами цілих невід'ємних чисел, поданих у вигляді монотонно зростаючих послідовностей, формуючи результат теж у вигляді монотонно зростаючої послідовності.

**Вхідні дані.** Вхідні дані завжди містять рівно 5 рядків.

1-ий: одне з трьох слів UNION або INTERSECTION або DIFFERENCE.

2-ий: єдине ціле число  $N$  ( $1 \leq N \leq 123456$ ), що задає кількість елементів множини  $A$ .

3-ій: послідовність з рівно  $N$  розділених одинарними пробілами чисел-елементів множини  $A$ ;  $0 \leq a_1 < a_2 < \dots < a_N \leq 10^9$ .

4-ий та 5-ий рядки задають множину  $B$  у такому ж форматі, як 2-ий і 3-ій множини  $A$ .

**Результати.** Виведіть у один рядок, розділяючи пробілами, усі елементи множини-відповіді. Рядок завершити символом переведення рядка.

Якщо результат не містить жодного елемента (наприклад, дія INTERSECTION, а  $A$  та  $B$  не мають спільних елементів), виведення

повинно не містити жодного видимого символу, але містити переведення рядка.

### Приклади:

Вхідні дані	Результати
UNION 3 2 3 5 3 1 2 4	1 2 3 4 5
INTERSECTION 3 2 3 5 3 1 2 4	2
DIFFERENCE 3 2 3 5 3 1 2 4	3 5

## Задача В. «Всюдисущі чісла»

Вхідні дані: Або клавіатура, або input.txt      Обмеження часу: 1 сек  
Результати: Або екран, або output.txt      Обмеження пам'яті: 64 мегабайти

Дано прямокутну таблицю  $N \times M$  чисел. Гарантовано, що у кожному окремо взятому рядку всі чісла різні й монотонно зростають.

Напишіть програму, яка шукатиме перелік (також у порядку зростання) всіх тих чисел, які зустрічаються в усіх  $N$  рядках.

**Вхідні дані.** У першому рядку задано два числа  $N$  та  $M$ . Далі йдуть  $N$  рядків, кожен з яких містить рівно  $M$  розділених пропусками чисел (гарантовано у порядку зростання).

**Результати.** Програма має вивести в один рядок через пробіли у порядку зростання всі ті чісла, які зустрілися абсолютно в усіх рядках. Кількість чисел виводити не треба. Після виведення всіх чисел потрібно зробити одне переведення рядка. Якщо нема жодного числа, що зустрілося в усіх рядках, виведення повинно не містити жодного видимого символу, але містити переведення рядка.

**Приклад:**

Вхідні дані	Результати
4 5	8 13
6 8 10 13 19	
8 9 13 16 19	
6 8 12 13 15	
3 8 13 17 19	

**Оцінювання.** У 20% тестів  $3 \leq N, M \leq 20$ , значення чисел від 0 до 100. У ще 20%,  $3 \leq N, M \leq 20$ , значення чисел від  $-10^9$  до  $+10^9$ . У ще 20%,  $1000 \leq N, M \leq 1234$ , значення від 0 до 12345. У решті 20%,  $1000 \leq N, M \leq 1234$ , значення від  $-10^9$  до  $+10^9$ .

Здавати потрібно одну програму, а не чотири; різні обмеження вказані, щоб пояснити, скільки балів можна отримати, розв'язавши задачу не повністю. (Враховуючи особливості саме цього туру, де досить великий штраф за хоча б один не пройдений тест, відсотки балів приблизно дорівнюють трьом чвертям згаданих відсотків тестів.)

## Задача С. «Школярі з хмарочосів»

Вхідні дані: Або клавіатура, або input.txt      Обмеження часу: 1 сек  
Результати: Або екран, або output.txt      Обмеження пам'яті: 64 мегабайти

У школі вчаться діти, які проживають у двох будинках–хмарочосах, розташованих поруч зі школою. Для того, щоб дійти до школи від 1-го хмарочосу, потрібно  $t_1$  часу, а від 2-го потрібно  $t_2$  часу. У 1-му хмарочосі живуть  $N_1$  школярів, у 2-му  $N_2$ . Про кожного школяра відомий час, коли він виходить з під'їзду.

Напишіть програму, яка з'ясовуватиме, в якому порядку вони приходитимуть до школи.

**Вхідні дані.** складаються з рівно шести рядків. Перший рядок містить єдине число — час, потрібний, щоб дійти від 1-го хмарочосу до школи. Другий рядок містить єдине число  $N_1$  — кількість школярів, що проживають у 1-му хмарочосі. У третьому рядку через пробіли записані (гарантовано впорядковані за строгим зростанням) моменти часу, коли школярі виходять із під'їзду. Рядки з четвертого по шостий описують, у такому самому форматі, учнів 2-го хмарочосу. Кількості учнів  $N_1$  та  $N_2$  можуть бути як однаковими, так і різними, кожна не менша 1 і не більша 98765. Усі значення часу є цілими числами у проміжку від 1 до 12345678.

**Результати.** Потрібно вивести перелік школярів в тому порядку, як вони приходять у школу. Дані кожного школяра мають бути виведені в окремому рядку, кожен такий рядок мусить мати вигляд: час, коли учень приходить до школи; номер хмарочоса, де він живе; номер, яким по порядку він виходить з під'їзду свого хмарочоса. Якщо різні учні приходять до школи одночасно (це можливо лише для учнів з різних хмарочосів), слід виводити спочатку дані про учня з 1-го хмарочоса, потім дані про учня з 2-го.

### Приклади:

Вхідні дані	Результати
10	10 1 1
3	14 1 2
0 4 7	15 2 1
15	17 1 3
2	18 2 2
0 3	

## Задача D. «Відстань між мінімумом та максимумом»

Вхідні дані: Або клавіатура, або input.txt      Обмеження часу: 1 сек  
Результати: Або екран, або output.txt      Обмеження пам'яті: 64 мегабайти

Напишіть програму, яка за заданим масивом цілих чисел знайде місце мінімального її елемента, максимального її елемента, та відстань (кількість проміжних елементів) між ними. У випадку, якщо масив містить однакові мінімальні та/або однакові максимальні елементи на різних позиціях, слід вибрати такий з мінімальних та такий з максимальних елементів, щоб шукана відстань виявилася мінімальною (див. також приклади 3 та 4).

**Вхідні дані.** Перший рядок містить єдине число  $N$  ( $2 \leq N \leq 10^6$ ) — кількість елементів масиву. Другий рядок містить (розділені пробілами) числа-елементи масиву; значення цих чисел-елементів цілі, від 1 до  $10^6$ .

**Результати.** Виведіть єдине ціле число — мінімально можлива відстань (кількість проміжних елементів) між якимсь із мінімальних і якимсь із максимальних елементів.

### Приклад:

Вхідні дані	Результати	Примітки
8 3 9 7 5 2 8 6 4	2	Між елементами 2 та 9 два проміжні елементи 5 та 7
8 4 3 2 1 8 7 6 5	0	Між елементами 1 та 8 нема проміжних елементів
8 3 1 4 1 5 9 2 6	1	Максимальний елемент (дев'ятка) єдиний; найближчою до єдиної дев'ятки є остання з мінімальних елементів (одиниць); між ними один проміжний елемент 5
8 1 1 1 1 1 1 1 1	0	Оскільки всі елементи однакові, кожен з них мінімальний і максимальний; між елементом і ним же самим нема проміжних

**Оцінювання.** У 20% тестів, масив містить єдине мінімальне та єдине максимальне значення; у 80% і мінімальне, і максимальне повторюються. Розподіл тестів за розмірами такий: 20% припадає на тести, де  $2 \leq N \leq 100$ ; 30% — на тести, де  $4000 < N \leq 10^4$ ; 20% — на тести, де  $5 \cdot 10^4 < N \leq 10^5$ ; 30% — на тести, де  $7 \cdot 10^5 \leq N \leq 10^6$ . Писати різні програми для різних випадків не треба; розподіли вказані, щоб показати, скільки приблизно балів можна отримати, розв'язавши задачу не повністю. (Враховуючи особливості саме цього туру, де досить великий штраф за хоча б один не пройдений тест, відсотки балів приблизно дорівнюють трьом чвертям згаданих відсотків тестів.)

## Задача Е. «Сума»

Вхідні дані: Або клавіатура, або input.txt

Обмеження часу: 1 сек

Результати: Або екран, або output.txt

Обмеження пам'яті: 64 мегабайти

Є гарантовано відсортована за строгим зростанням послідовність з  $n$  цілих чисел та число SUM\_NEED. («Строге» зростання означає, що кожен наступний строго більший за попередні, тобто не буває різних елементів з однаковим значенням.) Скільки є різних способів задати це число як суму  $a[i] + a[k]$ ? (Тобто, як суму з рівно двох доданків; допускається, якщо так виходить потрібна сума, щоб це був двічі взятий один елемент; способи  $a[i] + a[k]$  та  $a[k] + a[i]$  (при  $i \neq k$ ) вважаються різними.

**Вхідні дані.** Перший рядок містить кількість елементів послідовності  $n$  ( $1 \leq n \leq 123456$ ). Другий рядок містить  $n$  чисел — самі елементи. Третій рядок містить потрібну суму SUM\_NEED.

**Результати.** Єдине число — кількість способів отримати SUM\_NEED як  $a[i] + a[k]$ .

**Приклади:**

Вхідні дані	Результати
7 2 3 5 7 11 13 17 22	3

**Примітка.** Цими трьома способами є:

1.  $5 + 17$ ;
2.  $11 + 11$ ;
3.  $17 + 5$ .

## Задача F. «Хвіртка у паркані»

Вхідні дані: Або клавіатура, або input.txt      Обмеження часу: 1 сек  
Результати: Або екран, або output.txt      Обмеження пам'яті: 64 мегабайти

Пан Дивак вирішив оновити паркан, що відділяє його подвір'я від вулиці. Він вже закопав  $N$  стовпчиків, а потім згадав, що у паркані бажано б залишити хвіртку, шириною щонайменше  $W$ . Тепер йому, мабуть, доведеться викопувати деякі із вкопаних стовпчиків.

Щоб робота не була даремною, слід викопати якнайменше стовпчиків. Допоможіть панові Диваку визначити, скільки стовпчиків доведеться викопати. Після викопування стовпчиків мусить існувати проміжок (між двома залишеними стовпчиками, або між залишеним стовпчиком і одним із кінців ділянки, або між кінцями ділянки) ширини  $\geq W$ .

**Вхідні дані.** Перший рядок містить два цілих числа  $N$  та  $W$  — кількість вкопаних стовпчиків та мінімально необхідну ширину проміжку для хвіртки відповідно. Гарантується, що  $0 \leq N \leq 100000$ , і що  $0 \leq W \leq 10^9$ . Будемо вважати, що уздовж межі подвір'я введено вісь координат. У другому рядку вхідного файлу вказано два числа  $L$  та  $R$  ( $L < R$ ) — координати лівого і правого кінців межі подвір'я. Далі йде третій рядок, що містить  $N$  чисел — координати вкопаних стовпчиків. Усі координати (включаючи  $L$  та  $R$ ) — різні цілі числа, які не перевищують за модулем (абсолютною величиною)  $10^9$ . Гарантується, що всі стовпчики вкопані між лівим і правим кінцями.

**Результати.** Слід вивести єдине число у єдиному рядку — мінімальну кількість стовпчиків, які треба викопати. Якщо розв'язку не існує, то виведіть замість кількості число “-1” (без лапок).

**Приклади:**

Вхідні дані	Результати
3 2 2 6 3 4 5	1
3 2 1 6 4 3 5	0
3 5 1 7 5 3 4	3

**Примітка.** Гарантовано, що хоча б у половині тестів координати стовпчиків відсортовані за зростанням.

## Задача G. «Сума та різниця Мінковського»

Вхідні дані: Або клавіатура, або input.txt      Обмеження часу: 1 сек  
Результати: Або екран, або output.txt      Обмеження пам'яті: 64 мегабайти

Наведемо два різні формулювання однієї й тієї самої задачі.

---

Сума Мінковського двох фігур на координатній площині  $A, B$  — це фігура на координатній площині, якій належать ті й тільки ті точки, координати яких  $(x, y)$  можна подати як  $x = x_1 + x_2$ ,  $y = y_1 + y_2$ , де  $(x_1, y_1)$  — довільна з точок, що належать фігурі  $A$ , а  $(x_2, y_2)$  — довільна з точок, що належать фігурі  $B$ . Ця операція можлива для будь-яких фігур  $A, B$ .

Різниця Мінковського двох фігур на координатній площині  $A, B$  — це така фігура  $C$  на координатній площині, що фігура  $A$  є сумою Мінковського фігур  $B$  та  $C$ . Ця операція означена лише для деяких фігур  $A, B$  (а для решти пар  $A, B$  підібрати таку  $C$  неможливо).

Напишіть програму, яка читатиме описи фігур  $A, B$ , і знаходитиме суму та різницю Мінковського цих фігур. На вхід гарантовано подаватимуться лише фігури, котрі є опуклими багатокутками, причому одна зі сторін такого багатокутника горизонтальна, ліва вершина цієї сторони має координати  $(0, 0)$ , а решта сторін знаходяться у півплощині з додатними  $y$ .

Напишіть програму, котра вмітиме знаходити:

1. суму Мінковського;
2. різницю Мінковського

**Вхідні дані.** Спочатку 1 або 2 на позначення того, яку задачу потрібно розв'язувати ("1" — суму, "2" — різницю), потім описи опуклих багатокутників  $A$  та  $B$ , у такому форматі: спочатку число  $N$  ( $3 \leq N \leq 1000$ ) — кількість вершин у багатокутнику, а далі  $N$  груп по 2 цілих чисел  $x_i$  та  $y_i$  — координати вершин. Система координат завжди вибирається так, що перша вершина має координати  $(0; 0)$ , остання має координати  $(0; x_n)$ , де  $x_n > 0$ . Всі вхідні координати — цілі числа, що не перевищують за модулем (абсолютною величиною)  $10^6$  (деякі з  $x_i$  можуть бути від'ємними).

Всі числа записано в один рядок і розділено пропусками (пробілами).

**Результати.** Якщо розв'язується задача "2" і різниці для вказаних багатокутників не існує, то вивести на екран єдине число 0. Інакше вивести результат суми чи різниці Мінковського, в тому ж форматі, що для вхідних даних. Гарантовано, що використовуються лише такі вхідні дані, що якщо відповідь існує, то всі її координати є цілочисельними.

---

На станції Глухов-Товарний використовуються підйомні крани спеціальної конструкції "Мостовий-Глуховський". Крюк такого крану підвішений до кількох блоків, що їздять по рейці, розміщеній горизонтально (на певній висоті). Завдяки цьому, крюк можна переміщати в будь-яку точку частини площини, обмежену багатокутником спеціального вигляду: верхня сторона багатокутника збігається з рейкою крана, обидва внутрішні кути багатокутника при цій стороні гострі, решта вершин багатокутника розміщені довільним чином, але так, що багатокутник виявляється опуклим. Крім того, станція має в розпорядженні пристрій, котрий дозволяє комбінувати дію двох кранів такого типу: простір досяжності крюка скомбінованого механізму точно такий, якби рейку другого крана підвісили (зі збереженням горизонтальності) на крюк першого.

Напишіть програму, котра вмітиме виконувати такі дві дії:

---

1. За заданими областями досяжностей двох кранів знаходити область досяжності їхньої комбінації.
2. За заданими областю досяжності одного крану та потрібною областю досяжності, з'ясовувати, який потрібно взяти другий кран, щоб комбінація першого та другого кранів в точності співпадала з потрібною областю досяжності (або з'ясовувала, що це неможливо).

**Вхідні дані.** Спочатку 1 або 2 (на позначення того, яку задачу потрібно розв'язувати), потім йдуть дві області. Якщо розв'язується задача "1", то це області досяжності першого та другого кранів, якщо "2", то спочатку потрібна область досяжності, а потім область досяжності першого крану. Усі області досяжності задаються у такому форматі: спочатку число  $N$  ( $3 \leq N \leq 1000$ ) — кількість вершин у багатокутнику, а далі  $N$  груп по 2 цілих числа  $x_i$  та  $y_i$  — координати вершин цієї області в порядку зростання  $x$ -координати. Система координат завжди вибирається так, що перша вершина має координати  $(0; 0)$ , вісь  $y$  напрямлена згори донизу. Всі вхідні координати — цілі числа, що не перевищують за модулем (абсолютною величиною)  $10^6$ .

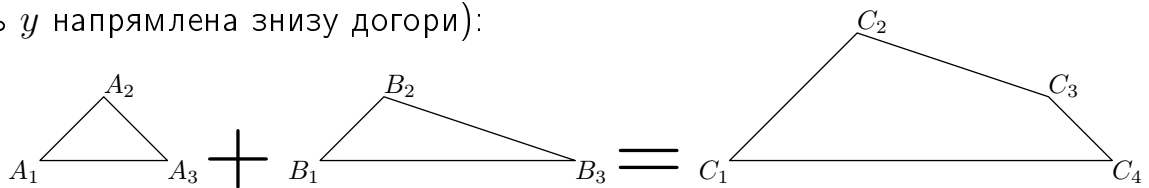
Всі числа записано в один рядок і розділено пропусками (пробілами).

**Результати.** Якщо розв'язується задача "2" і підібрати другий кран неможливо, то вивести на екран єдине число 0. Інакше вивести побудовану область досяжності (в тому ж форматі, що для вхідних даних). Гарантовано, що використовуються лише такі вхідні дані, що якщо відповідь існує, то всі її координати є цілочисельними.

**Приклади:**

Вхідні дані	Результати
2 5 0 0 10 40 20 50 30 40 40 0 3 0 0 10 10 20 0	3 0 0 10 40 20 0
2 3 0 0 10 10 20 0 3 0 0 10 10 40 0	0
1 3 0 0 10 10 20 0 3 0 0 10 10 40 0	4 0 0 20 20 50 10 60 0

**Примітка.** Пояснимо останній з наведених прикладів вхідних даних та результатів детальніше. Самі доданки та сума мають такий вигляд (за умови, що вісь  $y$  напрямлена знизу догори):



Цей результат може бути утворений, наприклад, внаслідок «ковзань» другого багатокутника уздовж сторін першого.

