

## Зміст

<b>1 Задачі основного дня (23.10.2016)</b>	<b>1</b>
А «Гра “Вгадай число”» . . . . .	1
Б «Бінпошук у масиві-1» . . . . .	3
В «Бінпошук у масиві-2» . . . . .	4
Г «Кубічне рівняння» . . . . .	6
Д «Дипломи» . . . . .	7
Е «Дошки» . . . . .	7
Ж «Всюдихід-1» . . . . .	8
И «Всюдихід-2» . . . . .	9

## 1 Задачі основного дня (23.10.2016)

Даний комплект задач доступний для on-line перевірки на сайті <http://ejudge.skipo.edu.ua/>, змагання № 53.

256 мегабайтів

### Задача 1. «Гра “Вгадай число”»

Напишіть програму, яка гратиме у гру «Вгадай число»: визначатиме загадане суперником ціле число із заданого діапазону на основі запитів до суперника.

У кожному запиті Ваша програма повинна виводити слово **try** та одне ціле число, що трактується як запитання «Загадане число . . . ?». Суперник на це відповідає (гарантовано чесно) одне з трьох: або, що саме це число й загадане, або, що загадане число більше, або, що загадане число менше. Ваша програма повинна продовжувати (або завершити) гру, враховуючи отримані відповіді. Кількість спроб вгадування (незалежно від величини проміжку) — до 50 (п'ятдесяти), включно.

При перевищенні цього ліміту, Вашій програмі присуджується технічна поразка (вона не отримує балів за відповідний тест).

Тобто, ця задача є інтерактивною: Ваша програма не отримає всіх вхідних даних на початку, а отримуватиме по мірі виконання доуточнення, котрі залежатимуть від попередніх дій Вашої програми. Тим не менш, її *перевірка теж відбувається автоматично*. Тому, у цій Вашій програмі, як і в програмах-розв’язках інших задач, теж слід *не* «організовувати діалог інтуїтивно зрозумілим чином», а чітко дотримуватися формату. Тільки це не формат вхідного та вихідного файлів, а формат спілкування з програмою, котра грає роль суперника.

### Вхідні дані

На початку, один раз, на вхід Вашій програмі подаються записані в один рядок через пропуск (пробіл) два цілі числа  $a, b$  ( $-2 \cdot 10^9 \leq a \leq b \leq 2 \cdot 10^9$ ) — межі проміжку. Це означає, що загадане ціле число гарантовано перебуває в межах  $a \leq x \leq b$ . На кожному наступному кроці, на вхід Вашій програмі подається рядок, що містить рівно один з трьох символів:

- `=` означає, що останнє виведене Вашою програмою число і є загаданим;
- `+` означає, що загадане число більше, ніж виведене Вашою програмою;
- `-` означає, що загадане число менше, ніж виведене Вашою програмою.

### Результати

Поки Ваша програма «не впевнена», яке число загадане, вона повинна повторювати виведення в один рядок через пропуск (пробіл або табуляцію) слова `“try”` (маленькими латинськими буквами, без лапок) та одного цілого числа — чергової спроби вгадування. Кожне число-спроба повинно бути в межах  $a \leq x \leq b$  (див. вище). Настійливо рекомендується після кожного такого виведення робити дію `flush(output)` (Pascal), вона ж `cout.flush()` (C++), вона ж `fflush(stdout)` (C), вона ж `sys.stdout.flush()` (Python), вона ж `System.out.flush()` (Java). Це істотно зменшує ризик, що проміжна відповідь «застряне» десь по дорозі, не дійшовши до програми-суперника.

Коли Вашій програмі «стає абсолютно очевидно», яке число загадане, вона повинна припинити основну частину гри і вивести в один рядок через пропуск (пробіл або табуляцію) слово “**answer**” (маленькими латинськими буквами, без лапок) та вгадане число. Після цього слід остаточно завершити роботу, не виводячи жодного іншого символу.

### Приклад

Вхідні дані	Результати	Примітки
1 100	try 10	1) Програма-суперник з самого початку загадала число 17.
+	try 20	2) Щоб було видніше, в якому порядку відбуваються введення та виведення, у прикладі використано розділення порожніми рядками. При фактичній перевірці програма-суперник не виводитиме ніяких порожніх рядків і не чекатиме їх від Вашої програми.
-	try 19	3) Ваша програма не зобов’язана грати за стратегією, що наведена у цьому прикладі. Це лише приклад можливої роботи програми, що успішно завершує конкретно цю гру.
-	try 18	
-	try 17	
=	answer 17	

256 мегабайтів

## Задача 2. «Бінпошук у масиві-1»

Дано два масиви. Гарантовано, що у першому з них всі елементи різні. Гарантовано, що другий містить лише елементи, які є в першому (але, можливо, не всі). Відсортуйте перший з них, а потім для кожного з елементів другого масиву знайдіть, під яким номером (нумерація починається з 1) він знаходиться у відсортованому (за зростанням) першому масиві.

### Вхідні дані

1-й рядок містить єдине число  $N$  ( $1 \leq N \leq 123456$ ) — кількість елементів 1-го масиву. 2-й рядок містить  $N$  розділених пропусками (пробіла-

ми) гарантовано різних чисел — елементи цього масиву. Не менш ніж 50% балів припадає на тести, де цей масив задано вже відсортованим (за зростанням), але у решті тестів це не так.

3-й рядок містить єдине число  $M$  ( $1 \leq M \leq N$ ) — кількість елементів 2-го масиву. 4-й рядок містить  $N$  розділених пропусками (пробілами) гарантовано різних чисел — елементи цього масиву. Кожен з елементів 2-го масиву гарантовано зустрічається також і в 1-му.

Значення елементів обох масивів є цілими числами, що не перевищують за модулем  $10^9$  (мільярд).

## Результати

Виведіть у один рядок через пропуски (пробіли) рівно  $M$  чисел: яким за номером у відсортованому вигляді 1-го масиву є 1-й елемент 2-го масиву, яким за номером у відсортованому вигляді 1-го масиву є 2-й елемент 2-го масиву, ..., яким за номером у відсортованому вигляді 1-го масиву є  $M$ -й елемент 2-го масиву. 2-й масив сортувати не слід, це повинні бути відповіді для того 2-го масиву, який задано у вхідних даних.

## Приклад

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
5 7 17 42 23 13 3 17 42 23	3 5 4

256 мегабайтів

## Задача 3. «Бінпошук у масиві–2»

Дано два масиви. Відсортуйте перший з них, а потім для кожного з елементів другого масиву знайдіть, чи зустрічається він у відсортованому першому масиві, і якщо зустрічається, то в якому діапазоні індексів (нумерація починається з 1).

## Вхідні дані

1-й рядок містить єдине число  $N$  ( $1 \leq N \leq 123456$ ) — кількість елементів 1-го масиву. 2-й рядок містить  $N$  розділених пропусками (пробілами) чисел — елементи цього масиву. Серед елементів можуть бути однакові. Не менш ніж третина балів припадає на тести, де цей масив задано вже відсортованим за неспаданням, але у решті тестів це не так.

3-й рядок містить єдине число  $M$  ( $1 \leq M \leq 123456$ ) — кількість елементів 2-го масиву. 4-й рядок містить  $M$  розділених пропусками (пробілами) чисел — елементи цього масиву.

Значення елементів обох масивів є цілими числами, що не перевищують за модулем  $10^9$  (мільярд).

## Результати

Виведіть у один рядок через пропуски (пробіли) рівно  $M$  рядків чисел, кожен з яких містить або єдине число 0 (якщо відповідне число 2-го масиву не зустрічається у 1-му масиві), або два розділені пробілом числа через пропуск (пробіл): індекс першого та індекс останнього входжень відповідного числа 2-го масиву у відсортований вигляд 1-го масиву. 2-й масив сортувати не слід, відповіді повинні бути для того 2-го масиву, який задано у вхідних даних.

## Приклад

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
7	4 6
17 7 17 42 7 17 7	7 7
3	0
17 42 23	

## Примітка

Відповідь 4 6 означає, що число 17 зустрічається на 4-й, 5-й та 6-й позиціях відсортованого вигляду 1-го масиву. Відповідь 7 7 — що число 42 зустрічається лише на 7-й позиції. Відповідь 0 — що число 23 взагалі не зустрічається.

256 мегабайтів

## Задача 4. «Кубічне рівняння»

Кубічне рівняння  $ax^3 + bx^2 + cx + d = 0$  задане чотирма своїми коефіцієнтами  $a, b, c, d$  ( $a \neq 0$ ). Як відомо, кубічне рівняння завжди має хоча б один дійсний корінь, а максимальна кількість різних дійсних коренів дорівнює 3. Напишіть програму, яка знайде усі дійсні корені кубічного рівняння.

### Вхідні дані

В одному рядку через пропуски (пробіли) задано чотири числа  $a, b, c, d$  — коефіцієнти рівняння. Усі чотири коефіцієнти є цілими числами, що не перевищують за модулем 1000, при цьому  $a \neq 0$  (решта коефіцієнтів можуть бути в тому числі й нулями).

### Результати

Виведіть в один рядок через пробіли (пропуски) усі дійсні корені рівняння. Якщо рівняння має менше, ніж три, корені, дозволяється деякі з них повторити, але так, щоб:

1. сумарна кількість виведених чисел була не більша трьох;
2. кожне з виведених чисел було коренем рівняння (дозволяється абсолютна та/або відносна похибка до  $10^{-6}$ );
3. кожен з коренів був виведений хоча б один раз.

### Приклади

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
1 -3 3 -1	1
1 -6 11 -6	1 2 3

### Примітка

Правила зарахування відповідей слід сприймати буквально. Наприклад, у першому прикладі зараховуються також відповіді, де 1 виведено два чи три рази. Більш того, зараховується також відповідь з трьома «різними» коренями 1, 1.000000001 та 0.999999999. А три різні корені 1, 1.001 та 0.999 вже не зараховуються, бо похибка перевищує дозволу. А чотири «різні» корені 1, 1.0000000001, 1.0000000002 та 0.999999999 не зараховуються не через похибки, а через те, що їх чотири, а дозволено максимум три.

256 мегабайтів

## Задача 5. «Дипломи»

Сергійко — дуже гарний олімпіадник. За його олімпіадну кар’єру в нього назбиралось досить багато дипломів. Тепер він випускник і хоче повісити всі дипломи на дошку, яку розмістить в своїй кімнаті. Він дуже хоче, щоб дошка була саме квадратною, також він знає ширину  $w$  та довжину  $h$  стандартного диплому. Допоможіть йому вибрати розмір дошки.

### Вхідні дані

В першому рядку знаходиться 3 числа  $w, h, n$  — ширина, довжина та кількість дипломів ( $1 \leq w, h, n \leq 10^9$ ).

### Результати

Виведіть єдине число — мінімально можливий розмір дошки, на яку можна повісити всі дипломи без накладань.

### Приклади

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
2 3 10	9
3 4 4	8

256 мегабайтів

## Задача 6. «Дошки»

(На жаль, на самій школі умова цієї задачі містила деякі неоднозначності, тому тепер вона трохи переписана.)

Степан вирішив оновити старий паркан біля дому. Для цього йому потрібно  $M$  дощок. Зараз у його сараї є  $N$  дощок однакової товщини та ширини, але різної довжини. Так як Степан любить засмагати влітку, він хоче, щоб паркан був якомога більшої однакової цілочисельної висоти. Степан просить допомогти йому і знайти таку найбільшу цілочисельну висоту. Дощки можна різати (як з метою приведення до

однакової довжини, так і з метою отримання з однієї дуже довгої кількох коротших), але не можна доточувати одну до одної.

### Вхідні дані

В першому рядку знаходяться два числа  $N, M$  ( $1 \leq N, M \leq 10000$ ). В кожному з наступних  $N$  рядків знаходиться по одному числу  $A_i$  — довжина  $i$ -ої дошки в сараї ( $1 \leq A_i \leq 10^7$ ).

### Результати

Виведіть єдине число — відповідь на задачу. Якщо Степан не може побудувати такий паркан, виведіть 0.

### Приклад

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
4 3 4 7 3 2	3

### Примітка

Три дошки висоті 3 можна отримати аж кількома різними способами: можна взяти дошки довжинами 3, 4, 7 і довші вкоротити до найкоротшої; можна взяти лише дошки довжинами 3 та 7, коротшу взяти цілою, а довшу розрізати на дві корисні по 3 і 1 у відході. Якби в умові не було вимоги про цілочисельність відповіді, то відповіддю було б 3,5 (довжину 7 навпіл плюс ще одна 3,5 із дошки довжини 4); але вимога цілочисельності є, тому 3.

256 мегабайтів

## Задача 7. «Всюдихід–1»

Всюдихід повинен виїхати з бази, перетнути спочатку пустелю, а потім болото і прибути на пост. Перешкод на шляху немає, всюдихід може рухатись у будь-якому напрямку. Максимальна швидкість всюдихода по пустелі і по болоту можуть відрізнятися одна від одної. Відомо,



що пряма, яка з'єднує базу і пост, проходить через обидві території. Визначте шлях, по якому всюдихід якнайшвидше прибуде на пост.

### Вхідні дані

1-й рядок містить максимальну швидкість всюдихода в пустелі  $v_1$  (м/с) та максимальну швидкість всюдихода по болоту  $v_2$  (м/с). 2-й рядок містить координати  $x_1$   $y_1$  бази. 3-й рядок містить координати  $x_2$   $y_2$  посту. Всередині кожного рядка числа розділені одним пропуском (пробілом). Відомо, що вісь  $Ox$  розділяє пустелю і болото (пустеля вгорі),  $y_1 > 0$ ,  $y_2 < 0$ ,  $x_1 \geq 0$ ,  $x_2 \geq 0$ . Всі числа дійсні і не перевищують за модулем  $10^7$ .

### Результати

Програма повинна вивести два дійсних числа — абсцису ( $x$ -координату) точки перетину межі територій і мінімальний час (у секундах), необхідний всюдиходу на поїздку від бази до посту. Відповідь зараховується, коли для кожного з цих двох чисел хоча б одна з похибок (абсолютна та/або відносна) не перевищує  $10^{-9}$ .

### Приклади

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
1 1 5 3 5 -2	5.0 5.0
3 5 20 10 8 -9	15.74651029 5.997280122

256 мегабайтів

## Задача 8. «Всудихід–2»

Всудихід має проїхати від старту  $S$  до фінішу  $F$  по степу, в якому є пустельна область. Пустельна область являє собою клин у вигляді частини площини між двома променями, що виходять з початку координат, один з променів спрямований угору вздовж осі  $Oy$ , інший перебуває у 1-й координатній чверті, утворюючи з першим кут  $\varphi$ . Пустелею всудихід може їхати з максимальною швидкістю  $v_1$ , степом — з макси-

мальною швидкістю  $v_0$  (причому  $v_0 \geq v_1$ ).

За який мінімальний час всюдихід може дістатися з точки  $S$  в точку  $F$ ?

### Вхідні дані

В єдиному рядку файлу через пропуски (пробіли) записано 7 дійсних чисел  $x_S, y_S, x_F, y_F, \varphi, v_0, v_1$ , котрі позначають: координати  $S(x_S, y_S)$ , координати  $F(x_F, y_F)$ , кут при вершині клина (в радіанах), швидкості всюдихода по степену і по пустелі.

Гарантовано, що:  $-1000 < x_S < 0 < x_F < 1000$ ;  $|y_S|, |y_F| < 1000$ ; обидві точки  $S$  та  $F$  перебувають поза пустелею і не на її межі; відрізок  $SF$  перетинає пустелю;  $\pi/180 < \varphi < \pi/4$ ;  $v_1 \leq v_0 \leq 4v_1$ ;  $0.1 \leq v_1 \leq 5$ .

### Результати

Вивести єдине дійсне число — шуканий мінімальний час. Виводити в будь-якому зі стандартних форматів для чисел з плаваючою крапкою (але крапкою, а не комою); відповідь зараховується, якщо її похибка (абсолютна або відносна, тобто хоча б одна з них) не більша  $10^{-6}$ .

### Приклад

stdin.txt або клавіатура (ст. вхід)	stdout.txt або екран (ст. вихід)
-2.0 4.0 2.5 5.0 0.3 3.5 2.0	1.6273547791