

Содержание

| | |
|--|----------|
| Must have | 2 |
| Задача 2А. Проверка ПСП [0.3 sec, 256 mb] | 2 |
| Задача 2В. Сумма простая [2.5 sec, 256 mb] | 3 |
| Задача 2С. Обмен [0.2 sec, 256 mb] | 4 |
| Обязательные задачи | 5 |
| Задача 2D. Хороша ли подстрока? [0.25 sec, 256 mb] | 5 |
| Задача 2Е. Художник [1.5 sec, 256 mb] | 6 |
| Задача 2F. Быстрое прибавление [5 sec, 256 mb] | 7 |
| Задача 2G. Сложение и вычитание [0.2 sec, 256 mb] | 8 |
| Дополнительные задачи | 9 |
| Задача 2H. Все минимумы [0.2 sec, 256 mb] | 9 |
| Задача 2I. Минимумы в подматрицах [0.4 sec, 256 mb] | 10 |
| Задача 2J. Game [1.5 sec, 256 mb] | 11 |
| Задача 2K. Интересный разбор выражений [2 sec, 256 mb] | 12 |

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

Must have

Задача 2А. Проверка ПСП [0.3 sec, 256 mb]

Дана строка, состоящая из круглых, квадратных и фигурных скобок. Нужно проверить, является ли она правильной скобочной последовательностью.

Формат входных данных

Во входном файле записана скобочная последовательность длиной не более 10 000 символов.

Формат выходных данных

Выведите YES, если скобочная последовательность является правильной, и NO в противном случае.

Примеры

| stdin | stdout |
|--------|--------|
| ([]()) | YES |
| ([]) | NO |

Задача 2В. Сумма простая [2.5 сек, 256 mb]

Вам нужно научиться отвечать на запрос “сумма чисел на отрезке”.
Массив не меняется. Запросов много. Отвечать на 1 запрос следует за $O(1)$.

Формат входных данных

Размер массива — n и числа x, y, a_0 , порождающие массив a : $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$. Далее следуют количество запросов m и числа z, t, b_0 , порождающие массив b : $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$, $c_i = b_i \bmod n$. i -й запрос — найти сумму на отрезке от $\min(c_{2i}, c_{2i+1})$ до $\max(c_{2i}, c_{2i+1})$ в массиве a .

Ограничения: $1 \leq n \leq 10^7, 0 \leq m \leq 10^7$. Все числа целые от 0 до 2^{16} . t может быть -1 .

Формат выходных данных

Выведите сумму всех сумм.

Пример

| stdin | stdout |
|----------|--------|
| 3 1 2 3 | 23 |
| 3 1 -1 4 | |

Замечание

$a = \{3, 5, 7\}, b = \{4, 3, 2, 1, 0, 2^{30} - 1\}, c = \{1, 0, 2, 1, 0, 0\}$,
запросы = $\{[0, 1], [1, 2], [0, 0]\}$, суммы = $\{8, 12, 3\}$.

Заметим, что вместо того, чтобы брать по модулю 2^{30} ,
достаточно всё считать в типе `int` и оставлять младшие 30 бит.

В Java есть знаковый сдвиг `>>` и беззнаковый `>>>`.

Одна из стандартных ошибок – переполнение типа. Проверьте, что у вас везде `int64`.

Задача 2С. Обмен [0.2 sec, 256 mb]

Пусть все натуральные числа исходно организованы в список в естественном порядке. Разрешается выполнить следующую операцию: $swap(a, b)$. Эта операция возвращает в качестве результата расстояние в текущем списке между числами a и b и меняет их местами.

Задана последовательность операций $swap$. Требуется вывести в выходной файл результат всех этих операций.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 200\,000$) — количество операций. Каждая из следующих n строк содержит по два числа в диапазоне от 1 до 10^9 — аргументы операций $swap$.

Формат выходных данных

Для каждой операции во входном файле выведите ее результат.

Пример

| stdin | stdout |
|-------|--------|
| 4 | 3 |
| 1 4 | 1 |
| 1 3 | 4 |
| 4 5 | 2 |
| 1 4 | |

Замечание

STL-структуры получают TL. Придётся писать свои.

Обязательные задачи

Задача 2D. Хороша ли подстрока? [0.25 сек, 256 mb]

Вам дана строка из круглых скобок. Нужно отвечать на запросы вида “есть ли у подстроки $[l..r]$ исходной строки циклический сдвиг, являющийся правильной скобочной последовательностью (ПСП)?”.

Напомним определение ПСП:

- Пустая строка – ПСП.
- Если s – ПСП, то $'(s)'$ – тоже.
- Если s_1, s_2 – ПСП, то $'s_1s_2'$ – тоже.

Формат входных данных

Строка s из круглых скобок длины от 1 до 10^6 .

На второй строке количество запросов q ($1 \leq q \leq 10^5$).

Следующие q строк содержат пары чисел l, r ($1 \leq l \leq r \leq |s|$).

Формат выходных данных

Для каждого запроса, если ответ да, выведите 1, иначе 0.

Примеры

| stdin | stdout |
|---|--------|
| ()())(6 1 2 1 4 5 6 3 6 1 5 2 5 | 111100 |

Задача 2Е. Художник [1.5 сек, 256 mb]

Енот Вася — начинающий художник. Недавно он приобрёл подержанную кисточку (как рассказал продавец, она сделана из хвоста самого Малевича!), достал холст и решил произвести на свет свой первый шедевр.

Как оказалось, лет кисточке немало, потому она способна лишь ставить кляксы в форме пяти квадратиков, расположенных крестиком (координаты их центров будут равны (x, y) , $(x-1, y)$, $(x, y-1)$, $(x+1, y)$, $(x, y+1)$). Вася поставил N клякс, разочаровался в идее первого шедевра и задумался о месте для нового. Но ведь если он закрасил весь холст, писать будет негде...

Выясните, закрасил ли своими действиями Вася весь холст.

Формат входных данных

Первая строка входного файла содержит три целых числа W , H и N — ширину и высоту холста в квадратиках ($1 \leq W, H \leq 10^9$) и количество клякс, поставленных Васей ($0 \leq N \leq 1\,000\,000$). Следующие N строк содержат по два целых числа x_i, y_i каждая — координаты среднего квадратика i -й кляксы ($-10^9 \leq x_i, y_i \leq 10^9$). Клетка (x, y) находится на холсте, если $1 \leq x \leq W$ и $1 \leq y \leq H$. Части клякс, оказавшиеся вне холста, не учитываются.

Формат выходных данных

Выведите «Yes», если Вася закрасил весь холст, и «No» в противном случае.

Пример

| stdin | stdout |
|---------------------|--------|
| 2 2 2 1 1 2 2 | Yes |

Задача 2F. Быстрое прибавление [5 сек, 256 mb]

Есть массив целых чисел длины $n = 2^{24}$, изначально заполненных нулями. Вам нужно сперва обработать m случайных запросов вида “прибавление на отрезке”. Затем обработать q случайных запросов вида “сумма на отрезке”.

Формат входных данных

На первой строке числа m, q ($1 \leq m, q \leq 2^{24}$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
0. unsigned int a, b; // даны во входных данных
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
```

Каждый запрос первого вида генерируется следующим образом:

```
1. add = nextRand(); // число, которое нужно прибавить
2. l = nextRand();
3. r = nextRand();
4. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Каждый запрос второго вида генерируется следующим образом:

```
1. l = nextRand();
2. r = nextRand();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Сперва генерируются запросы первого вида, затем второго.

Формат выходных данных

Выведите сумму ответов на все запросы второго типа по модулю 2^{32} .

Примеры

| stdin | stdout |
|---------------|-----------|
| 5 5 13 239 | 811747796 |

Замечание

Последовательность запросов в тесте из примера:

```
[13..170] += 0
[28886..375523] += 2221
[2940943..13131777] += 4881801
[2025901..10480279] += 4677840
[4943766..6833065] += 9559505
get sum [13412991..13937319]
get sum [1871500..6596736]
get sum [7552290..14293694]
get sum [1268651..16492476]
get sum [2210673..13075602]
```

Задача 2G. Сложение и вычитание [0.2 сек, 256 mb]

Выведите значение заданного арифметического выражения, состоящего из чисел, скобок и знаков сложения и вычитания.

Формат входных данных

В первой строке входного файла задано выражение, состоящее из чисел, скобок и знаков бинарных операций. Каждое число в выражении это — целое неотрицательное число в промежутке от 0 до 10 000, включительно, записанное без ведущих нулей. Скобки бывают открывающие ('(') и закрывающие (')'). Операции задаются символами '+' и '-'. Гарантируется, что заданное выражение математически корректно, и результаты всех промежуточных операций — целые числа, не превышающие по модулю 10 000. Выражение не содержит каких-либо других символов, в частности, пробелов. Длина выражения не меньше 1 и не больше 1000 символов.

Учтите, что операции при отсутствии скобок выполняются слева направо. Например, выражение $a - b - c$ вычисляется как $(a - b) - c$.

Формат выходных данных

В первой строке выходного файла выведите одно число — значение заданного выражения.

Примеры

| stdin | stdout |
|----------|--------|
| 48-13 | 35 |
| 5-(52+3) | -50 |

Дополнительные задачи

Задача 2Н. Все минимумы [0.2 сек, 256 mb]

Дан массив целых чисел a_1, a_2, \dots, a_n .

Для каждого его подотрезка $[a_L..a_R]$ определим $F(L, R) := \min\{a_L, \dots, a_R\}$.

Найдите

$$\sum_{1 \leq L \leq R \leq n} F(L, R)$$

то есть сумму минимумов всех подотрезков.

Формат входных данных

Первая строка входных данных содержит натуральное число n ($1 \leq n \leq 100\,000$) — размер массива. Во второй строке через пробел заданы элементы массива, все числа целые от -10^6 до 10^6 .

Формат выходных данных

Выведите единственное число — сумму минимумов всех подотрезков массива a .

Примеры

| stdin | stdout |
|-------------------|--------|
| 1 5 | 5 |
| 2 -10 1 | -19 |
| 4 1 2 3 4 | 20 |
| 5 -3 2 -4 1 -5 | -52 |

Замечание

Принимаются только решения за $\mathcal{O}(n)$. ОК-сабмиты будут подвергнуты code review.

Задача 21. Минимумы в подматрицах [0.4 sec, 256 mb]

Дана матрица $n \times n$, состоящая из целых чисел. Для каждой её подматрицы размера $L \times L$ найдите минимум в этой подматрице. Подматрицей здесь называется “подпрямоугольник”.

Внимание. Решение должно работать за $O(n^2)$.

Формат входных данных

Первая строка входных данных содержит два целых числа n и L ($1 \leq L \leq n \leq 1000$). Далее в n строках идет описание матрицы $n \times n$, по n чисел в каждой строке. Все числа в матрицы целые, от -10^9 до 10^9 .

Формат выходных данных

Выведите $n - L + 1$ строку по $n - L + 1$ числу в каждой. j -е число в i -й строке должно быть равно минимуму в подматрице размера $L \times L$ с левым верхним углом на пересечении i -й строки и j -го столбца исходной матрицы.

Примеры

| stdin | stdout |
|---|-------------------------|
| 1 1 5 | 5 |
| 2 1 2 1 3 4 | 2 1 3 4 |
| 2 2 2 1 3 4 | 1 |
| 4 2 4 5 3 2 1 2 5 4 3 4 2 3 1 3 5 5 | 1 2 2 1 2 2 1 2 2 |

Задача 2J. Game [1.5 sec, 256 mb]

Вася и Петя снова затеяли очень интересную игру! Петя выписывает на доску по порядку n чисел. Теперь мальчики, не советуясь, выбирают по 2 числа от 1 до n .

Пусть Петя выбрал числа l и r ($l \leq r$). Тогда очки, заработанные им, вычисляются по формуле $\sum_{k=l}^r b_k \cdot \min_{k=l..r} b_k$, где b_k — k -ое число на доске. Аналогично вычисляются очки, заработанные Васей. Затем мальчики говорят друг другу свои очки. У кого очков меньше, тот проиграл. Если очков одинаковое количество, то это считается ничьей.

Представьте, что вы — Вася. И вы очень хотите хотя бы не проиграть. При этом очень не хотите быть пойманными на жульничестве (то есть хотите, чтобы названное вами число очков возможно было получить честным способом).

Напишите программу, которая находит отрезок, дающий максимальное количество очков.

Формат входных данных

В первой строке ввода задано натуральное число n ($1 \leq n \leq 2\,000\,000$) — количество чисел на доске. В следующей строке содержатся n целых чисел b_k через пробел в том же порядке, что они написаны на доске ($|b_k| \leq 100\,000$).

Формат выходных данных

Выведите число, которое должен назвать Вася, чтобы заведомо не проиграть и не быть пойманным на жульничестве. Во второй строке выведите границы отрезка, чтобы Вася смог доказать свою честность, если его начнут подозревать в жульничестве. Удачи!

Пример

| stdin | stdout |
|-------|--------|
| 3 | 10 |
| 1 2 3 | 2 3 |

Задача 2К. Интересный разбор выражений [2 sec, 256 mb]

Задача: дано арифметическое выражение, посчитайте значение.

В выражении присутствуют:

- Целые числа из диапазона $[-2^{31}, 2^{31})$.
- Операции:
 - + , - , * (сложение, вычитание, умножение),
 - % (остаток по модулю, не отрицателен),
 - / (целочисленное деление, остаток неотрицателен),
 - ^ (возведение в степень).
- Унарный минус.
- Скобки трех типов: { } [] ().
- Переменные с целочисленными значениями. Имена переменных — строки из букв латинского алфавита. Регистр важен.
- Функции `sq` (квадрат числа), `cube` (куб числа), `sign` (знак числа).

Правила разбора выражения:

- Минус: после числа/имени или закрывающей скобки идет бинарный минус. Иначе минус унарный.
- Приоритеты операций: (+, -) затем (*, %, /) затем (^).
- Операции +, -, *, / левоассоциативны: $2-3+4 = (2-3)+4$.
- Операция возведения в степень ^ правоассоциативна: $3^3^3 = 3^{27}$.
- Если после имени идет открывающая скобка — это функция, иначе переменная.

Вычисление выражения: сперва происходит разбиение на лексемы и построение дерева вычислений, затем вычисляется значение выражения с помощью обхода дерева разбора слева направо. Сперва вычисляются значения аргументов операции в порядке слева направо, а когда все аргументы посчитаны, вычисляется значение оператора. Унарные операторы и функции в дереве имеют степень один, бинарные операторы имеют степень два.

Формат входных данных

Первые несколько строк содержат значения переменных в формате $\langle \text{name} \rangle = \langle \text{value} \rangle$. Эти строки точно корректны. Имена переменных в присваиваниях могут совпадать, используется последнее значение. Последняя строка содержит арифметическое выражение, значение которого нужно посчитать. Арифметическое выражение может содержать синтаксические ошибки, и ошибки, не позволяющие вычислить его значение. Подробнее читайте ниже. Суммарная длина всех строк входного не более 10^6 . Во входном файле используются только допустимые символы.

Формат выходных данных

Если арифметическое выражение некорректно, выведите `Error: <ошибка>`. Если произошло несколько ошибок, нужно выводить только первую.

Ошибки на этапе разбора выражения на лексемы в порядке приоритета:

- `unmatched bracket` — лишние или не парные скобки.
- `parsing expression` — численные значения и операции не чередуются.
- `undefined name` — имя, которое не соответствует ни переменной, ни функции.
- `too long integer` — используется число не из диапазона $[-2^{31}, 2^{31})$.

Ошибки на этапе вычисления значения выражения, выражение вычисляется обходом дерева слева направо, нас интересует первая ошибка именно в этом порядке:

- `integer overflow` — конечное, или одно из промежуточных значений лежат за пределами диапазона $[-2^{31}, 2^{31})$.
- `dividing by zero` — деление на ноль.
- `negative power is not allowed` — возведение в отрицательную степень.

Если арифметическое выражение задано корректно и может быть корректно вычислено, выведите целое число — значение выражения.

Примеры

| stdin | stdout |
|---|---|
| <code>x = 2 value = 3 x^value - cube(2+3)</code> | <code>-117</code> |
| <code>-2+3-(-2+3)+[-100]-{-100}--100+ sqr(-2)+sign(-1)+-2^5</code> | <code>71</code> |
| <code>x = 2 x^100</code> | <code>Error: integer overflow</code> |
| <code>()</code> | <code>Error: parsing expression</code> |
| <code>(2 2 / 0 + 1000000000000</code> | <code>Error: unmatched bracket</code> |
| <code>(2 2) / 0 + 1000000000000</code> | <code>Error: parsing expression</code> |
| <code>(2 + x) / 0 + 1000000000000</code> | <code>Error: undefined name</code> |
| <code>(2 + 2) / 0 + 1000000000000</code> | <code>Error: too long integer</code> |
| <code>(2 + 2) / 0</code> | <code>Error: dividing by zero</code> |
| <code>2^(2 - sqr(2))</code> | <code>Error: negative power is not allowed</code> |
| <code>sqr = 8 sqr = 10 sqr + (sqr + 2 + sqr) + sqr(sqr(sqr)) + sqr</code> | <code>10042</code> |