

### House36. Тридцать шестой домик

Имя входного файла: house36.in  
Имя выходного файла: house36.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Тридцать шестой домик в «Берендеевых полянах» состоит из бесконечного количества одноместных комнат, пронумерованных натуральными числами. В этот домик регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он селится в первую свободную комнату с большим номером.

Кроме того, некоторые дети уезжают из домика в середине смены. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу преподавателя, ответственного за этот домик, и научитесь быстро сообщать приезжающим детям, в какую комнату им следует селиться.

#### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество прибытий и отъездов, происходящих в течение смены ( $n \leq 100\,000$ ).

Следующие  $n$  строк содержат информацию об ЛКШтатах. Число  $a > 0$  обозначает, что приехал школьник, желающий заселиться в комнату номер  $a$  ( $a \leq 100\,000$ ). Число  $a < 0$  обозначает, что из комнаты номер  $-a$  уехал школьник. (Гарантируется, что эта комната не была пуста).

#### Формат выходного файла

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

#### Пример

| house36.in | house36.out |
|------------|-------------|
| 6          | 5           |
| 5          | 6           |
| 5          | 7           |
| 5          | 6           |
| -6         | 8           |
| 5          |             |
| 5          |             |

### RVQ. Range Variation Query

Имя входного файла: rvq.in  
Имя выходного файла: rvq.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В начальный момент времени последовательность  $a_n$  задана следующей формулой:

$$a_n = n^2 \bmod 12345 + n^3 \bmod 23456$$

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значением среди элементов  $a_i, a_{i+1}, \dots, a_j$ ;
- присвоить элементу  $a_i$  значение  $j$ .

#### Формат входного файла

Первая строка входного файла содержит натуральное число  $k$  — количество запросов ( $k \leq 100\,000$ ). Следующие  $k$  строк содержат запросы, по одному на строке. Запрос номер  $i$  описывается двумя целыми числами  $x_i, y_i$ .

Если  $x_i > 0$ , то требуется найти разность между максимальным и минимальным значением среди элементов  $a_{x_i} \dots a_{y_i}$ . При этом  $1 \leq x_i \leq y_i \leq 100\,000$ .

Если  $x_i < 0$ , то требуется присвоить элементу  $a_{-x_i}$  значение  $y_i$ . При этом  $-100\,000 \leq x_i \leq -1$  и  $|y_i| \leq 100\,000$ .

#### Формат выходного файла

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значением на соответствующем отрезке.

#### Пример

| rvq.in  | rvq.out |
|---------|---------|
| 7       | 34      |
| 1 3     | 68      |
| 2 4     | 250     |
| -2 -100 | 234     |
| 1 5     | 1       |
| 8 9     |         |
| -3 -101 |         |
| 2 3     |         |

### Шифр Бэкона

Имя входного файла: bacon.in  
Имя выходного файла: bacon.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Программисту Васе не повезло — вместо отпуска его послали в командировку, на научную конференцию. Надо повышать уровень знаний, сказал начальник, важная конференция по криптографии, проводится во Франции — а там шифровали еще во времена Рипелье и взламывали чужие шифры еще во времена Виета. Вася быстро выяснил, что все луврские картины он уже где-то видел, вид Эйфелевой башни приелся ему еще раньше, чем мышка стерла его с коврика, а такие стеклянные пирамиды у нас делают надо всякими киосками и сомнительными забегаловками. Одним словом,

смотреть в Париже оказалось просто не на что, рыбу половить негде, поэтому Васе пришлось посещать доклады на конференции. Один из докладчиков, в очередной раз пытаясь разгадать шифры Бэкона, выдвинул гипотезу, что ключ к тайнам Бэкона можно подобрать, проанализировав все возможные подстроки произведений Бэкона. «Но их же слишком много!» — вслух удивился Вася. «Нет, не так уж и много!» — закричал докладчик — «подсчитайте и вы сами убедитесь!». Тем же вечером Вася нашел в интернете полное собрание сочинений Бэкона. Он написал программу, которая переработала тексты в одну длинную строку, выкинув из текстов все пробелы и знаки препинания. И вот теперь Вася весьма озадачен — а как же подсчитать количество различных подстрок этой строки?

#### **Формат входного файла**

На входе дана непустая строка, полученная Васей. Строка состоит только из строчных латинских символов. Ее длина не превосходит 2000 символов.

#### **Формат выходного файла**

Выведите количество различных подстрок этой строки.

#### **Пример**

| bacon.in | bacon.out |
|----------|-----------|
| aaba     | 8         |