

Kody pocztowe (kody-pocztowe)

Limit pamięci: 64 MB

Limit czasu: 0.50 s

Jasio, przechodząc przez miasto zauważył na murze graffiti. Nie było ono typowym kolorowym murałem lub bezsensownym tagiem autora. Na murze napisany jest bowiem ciąg cyfr i myślników. Jasio zastanawiał się jaki sens może mieć ten zapis. Czy chodzi o działanie odejmowania? Czy chodzi o liczby pooddzielane myślnikami? W końcu wymyślił! Na pewno autor chciał schować w graffiti kod pocztowy. Dla przypomnienia: format kodu pocztowego jest następujący $[0-9][0-9]-[0-9][0-9][0-9]$, czyli dwie cyfry, następnie znak myślnika, następnie trzy cyfry. Przykładowo, 00-789 oraz 23-456 to poprawne kody pocztowe, zaś 12345, 123-45, 00-0001, 0-1234 nie są poprawne. Jasio zastanawia się teraz ile różnych kodów pocztowych może uzyskać, jeżeli by zamazać niektóre (być może żadnej) znaki, a pozostałe odczytać od lewej do prawej. Pomóż mu w tym zadaniu.

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się niepusty ciąg cyfr oraz znaków -.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita – liczba różnych kodów pocztowych jakie można odnaleźć w napisie na wejściu.

Ograniczenia

Długość napisu na wejściu nie przekracza 100 000 znaków.

Przykład

Wejście

2111-3-411

Wyjście

8

Wyjaśnienie

W tym przypadku pasują następujące kody: 11-311, 11-341, 11-411, 13-411, 21-311, 21-341, 21-411, 23-411.

Maszyna sumująco-zwiększająca (maszyna-sum-zwi)

Limit pamięci: 64 MB

Limit czasu: 2.00 s

Jasio, po ukończeniu studiów na politechnice, jest inżynierem w zakładzie produkcyjnym maszyn liczbowych. Ma teraz bardzo ważne zadanie: zaprojektować maszynę sumująco-zwiększającą.

Maszyna ta powinna umożliwiać następujące operacje:

- INSERT x_i – wstaw liczbę x_i do środka maszyny,
- INCREASE d_i – zwiększ każdą liczbę, która jest obecnie w maszynie o d_i ,
- SUM – podaj sumę wszystkich liczb znajdujących się obecnie w maszynie.

Zanim inni inżynierowie i pracownicy w zakładzie zaczną przygotowywać wielkie maszyny realizujące te ważne zadania, Jasio musi przemyśleć jak te maszyny będą działać i przygotować tak zwany *proof-of-concept*, czyli symulator działania gotowej maszyny. Postanowiono, że będzie to program komputerowy. Jasio niestety nie za dobrze programuje, na politechnice nauczył się głównie rysunku technicznego i geometrii wykreślnej, dlatego poprosił Cię o pomoc.

Napisz program, który: wczyta operacje do maszyny sumująco-zwiększającej, wyznaczy i wypisze wyniki na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna Q , określająca liczbę operacji. W kolejnych Q wierszach znajduje się opis kolejnych operacji: jedno ze słów INSERT, INCREASE lub SUM oraz:

- w przypadku INSERT: pojedynczy odstęp oraz liczba naturalna x_i ,
- w przypadku INCREASE: pojedynczy odstęp oraz liczba naturalna d_i .

Wyjście

Twój program powinien wypisać odpowiedzi na kolejne zapytania SUM w kolejnych wierszach.

Ograniczenia

$1 \leq Q \leq 500\,000$, $1 \leq x_i \leq 1\,000\,000$, $1 \leq d_i \leq 1\,000\,000$.

Przykład

Wejście

```
7
INSERT 3
INSERT 7
INSERT 3
INCREASE 2
SUM
INSERT 3
SUM
```

Wyjście

```
19
22
```

Wyjaśnienie

Po pierwszych trzech operacjach w maszynie znajdują się liczby $\{3, 3, 7\}$. Po czwartej operacji w maszynie znajdują się liczby $\{5, 5, 9\}$. Ich suma wynosi 19. Następnie, po kolejnej operacji INSERT w maszynie znajdują się liczby $\{5, 5, 9, 3\}$. Ich suma wynosi 22.

Wyrównywanie liczb (wyrownywanie-liczb)

Limit pamięci: 64 MB

Limit czasu: 0.75 s

Dane są dwie liczby naturalne A oraz B . Naszym celem jest w tym zadaniu jest spowodować, żeby $A = B$. Możliwe jest wykonywanie następujących operacji:

- $A \leftarrow s(A)$,
- $B \leftarrow s(B)$,
- $A \leftarrow A + 1$,
- $B \leftarrow B + 1$.

Funkcja $s(\cdot)$ jako argument przyjmuje liczbę naturalną i zwraca jako wynik jej sumę cyfr. Przykładowo: $s(123) = 6$.

Ile najmniej operacji należy wykonać, żeby spowodować, że podane na wejściu liczby będą równe? Żeby nie było tak łatwo, Twój program będzie musiał rozpatrzyć wiele przypadków par (A_i, B_i) i dla każdej z nich (szybko) udzielić poprawnej odpowiedzi.

Napisz program, który: wczyta liczbę zapytań oraz dla każdego z nich liczby naturalne A_i i B_i , odpowie na wszystkie zapytania, tj. wyznaczy minimalną liczbę operacji, które należy wykonać, żeby wyrównać liczby A_i i B_i i wypisze wyniki na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna Q , określająca liczbę zestawów danych. W kolejnych Q wierszach znajduje się opis kolejnych zestawów danych, po jednym w wierszu. Opis każdego zestawu danych składa się z dwóch liczb naturalnych A_i oraz B_i , oddzielonych pojedynczym odstępem.

Wyjście

Twój program powinien wypisać dokładnie Q wierszy. W i -tym z nich powinna się znaleźć odpowiedź dla i -tego zestawu danych, czyli jedna nieujemna liczba całkowita – minimalna liczba operacji, które należy wykonać, aby spowodować, że liczby A_i oraz B_i będą sobie równe.

Ograniczenia

$$1 \leq Q \leq 100\,000, 0 \leq A_i, B_i \leq 10^{18}.$$

Przykład

Wejście

4
12 18
1 9
25 7
58 71

Wyjście

4
2
1
6
6

Wyjaśnienie

W pierwszym przypadku możemy wykonać następujący ciąg operacji:
 $(12, 18) \rightarrow (12, 19) \rightarrow (12, 10) \rightarrow (12, 11) \rightarrow (12, 12)$.