|  |  |
| --- | --- |
| *Autor zadania: Jakub Bachurski* | 6 lipca 2018 (wersja 1.3.2) |

|  |  |
| --- | --- |
| Dostępna pamięć: 64 MB | Limit czasu: 4 sekundy |

*Bity*

Cinomeusz i Coutolomeo, podczas ekspedycji do pewnych podziemnych lochów, znaleźli antyczną maszynę podobną do dzisiejszego komputera. Wykorzystuje ona system dwójkowy. Wygląda jednak na to, że nie mogą jeszcze z niej skorzystać – aby ją odblokować będą musieli wykonywać operacje bitowe (dokładniej AND, OR oraz XOR z daną liczbą) na przedziałach na danym ciągu liczb, oraz podawać sumę na przedziale. Zapytań, rzecz jasna, jest zdecydowanie za dużo, by poradzili sobie sami z tym zadaniem. Czy im pomożesz?

# Wejście

Na pierwszym wierszu wejścia znajdują się dwie liczby: oraz (. W kolejnym wierszu znajduje się liczb całkowitych ( oddzielonych spacjami – kolejne liczby w ciągu. Ciąg jest indeksowany od 0. W kolejnych wierszach znajdują się zapytania postaci „c a b” (znak – odpowiednio operacja AND, OR, XOR oraz zapytanie o sumę; liczby całkowite , oznaczające przedział: ). Dodatkowo, jeżeli , w tym samym wierszu pojawi się jeszcze liczba całkowita (), z którą zostanie wykonana operacja bitowa.

Przykładowo, gdy pojawia się wiersz „^ 1 3 5”, twoim zadaniem jest XOR-ować liczby na indeksach 1, 2, 3 z liczbą 5. Gdy pojawia się wiersz „? 0 3”, mamy wypisać sumę liczb na indeksach   
0, 1, 2 oraz 3.

# Wyjście

Dla każdego zapytania na wyjście należy wypisać jeden wiersz zawierający jedną liczbę całkowitą, oznaczającą sumę wszystkich liczb na przedziale, o który było zapytanie.

# Przykłady

|  |  |
| --- | --- |
| Wejście | Wyjście |
| 5 5  1 4 3 2 6  ^ 2 3 4  ? 1 4  & 1 2 3  | 0 4 1  ? 0 4 | 23  19 |

# Wyjaśnienie do przykładu

|  |  |  |  |
| --- | --- | --- | --- |
| **Zapytanie** | **Ciąg (po operacji)** | **Wynik** | **Wyjaśnienie** |
| - | 1 4 3 2 6 | - |  |
| ^ 2 3 4 | 1 4 **7 6** 6 | - | 7 = 3^4, 6 = 2^4 |
| ? 1 4 | 1 **4 7 6 6** | 23 | = 4 + 7 + 6 + 6 |
| & 1 2 3 | 1 **0 3** 6 6 | - | 0 = 4&3, 3 = 7&3 |
| | 0 4 1 | **1 1 3 7 7** | - | 1 = 1|1, 1 = 0|1, 3 = 3|1,  7 = 6|1, 7 = 6|1 |
| ? 0 4 | **1 1 3 7 7** | 19 | = 1 + 1 + 3 + 3 + 7 |

Nie, O(nq) ma nie przechodzić.