**Problem A-Цукерки**

**Цукерки**

|  |  |
| --- | --- |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Петрик та Василь дуже люблять цукерки.Та вони їх не їдять,а колекціонують.Петрик полюбляє цукерки з малюнком ведмедя на обгортці, а Василь полюбляє з малюнком лисиці.У Петрика є непотрібні йому цукерки (обгортки цукерок не подобаються Петрикові). І у Василька є непотрібні йому цукерки. Також Петрик і Василь полюбляють обмінюватись непотрібними цукерками. Вони обмінюють всі свої непотрібні , на всі непотрібні цукерки товарища.

Вхідні дані:
Дано 2 числа ***N, K (1 ≤ N, K ≤ 109)*** кількість непотрібних цукерок відповідно Петрика і Василька.

Вихідні дані:
Oдне число кількість цукерок, які отримав Петрик під час обміну.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 | 2 |
| 2 3 | 3 |

**Problem B-Звичайна арифметика**

**Звичайна арифметика**

|  |  |
| --- | --- |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Дано два числа (a і b) та операція (словом), котру необхідно виконати:
**plus** – додати ці числа
**minus** – відняти друге число від першого
**div** - знайти цілу частину від ділення першого числа на друге
**mod** – знайти остачу від ділення першого числа на друге
**mult** – перемножити дані числа
Напишіть програму, яка за даними числами та операцією, обчислює результат.

Вхідні дані:
Дано 2 числа ***a, b (-2\*109 ≤ a, b ≤ 2\*109)*** та операцію, яка записана через один пробіл.

Вихідні дані:
Oдне число - результат виконаної операції.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 4 plus | 7 |
| 12 5 div | 2 |
| 9 4 mod | 1 |
| 4 5 minus | -1 |
| 7 3 mult | 21 |

**Problem C-Шкідлива речовина**

**Шкідлива речовина**

|  |  |
| --- | --- |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

В одному з боксів хімічної лабораторії, де ведуться наукові дослідження, відбувся витік шкідливої речовини. Коли ця речовина сягає зовнішніх стін лабораторії, вмикається попереджувальна сирена.
Відомо, що лабораторія складається з ***n*** боксів по горизонталі та ***m*** боксів по вертикалі, які мають нумерацію ***(i, j)***, де ***i*** – номер рядка, а ***j*** – номер боксу в рядку ***(1 ≤ i ≤ n, 1 ≤ j ≤ m)***. Витік шкідливої речовини відбувся у боксі з номером ***(х, у)*** і протягом 1 хвилини поширюється у сусідні бокси через спільні з ним стіни.

Вхідні дані:
у першому рядку містяться розміри лабораторії ***n, m (1 ≤ n, m ≤ 109)***, у другому – нумерація боксу ***x, y (1 ≤ x ≤ n, 1 ≤ y ≤ m)***, де відбувся витік шкідливої речовини.

Вихідні дані:
одне число, яке визначає мінімальну кількість хвилин, через яку увімкнеться попереджувальна сирена.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 52 3 | 1 |
| 5 53 3 | 2 |

**Problem D-Гра**

**Гра**

|  |  |
| --- | --- |
| **Time limit:** | 500 ms |
| **Memory limit:** | 256 M |

У парку встановили новий атракціон, який був зроблений у вигляді закритого круглого приміщення, куди заходив гравець. У приміщенні вимикалося світло і по колу на стінах з’являлися білі і чорні квадрати. Гравець за один крок гри може прибирати два квадрати однакового кольору, притискаючи до них свої долоні. Послідовність прибирання квадратів різних кольорів немає значення, тобто спочатку можна прибрати всі квадрати одного кольору, а потім іншого, або у будь-якій іншій послідовності.
Гравець виграє приз, якщо у результаті залишиться один квадрат білого кольору, і залишиться без призу, якщо залишиться лише один квадрат чорного кольору. У разі, коли не залишиться жодного квадрату, то у нього з’являється можливість зіграти ще раз.
Допоможіть гравцю якнайшвидше визначитися з перспективами 5 сеансів гри.

Вхідні дані:
у першому рядку міститься число ***w***, яке визначає початкову кількість квадратів білого кольору в першій грі, у другому рядку – число ***b***, яке визначає початкову кількість квадратів чорного кольору в першій грі ***(1 ≤ w, b ≤ 101000000)***. У третьому та четвертому рядках – числа ***w*** та ***b***, що визначають початкову кількість квадратів відповідно білого та чорного кольорів в другій грі. Аналогічно для решти (всього 10 рядків – по 2 на кожну з 5 ігор).

Вихідні дані:
для кожної з 5 ігор в окремому рядку вивести ‘Lose’, якщо гравець не отримає приз, ‘Win’, якщо гравець отримає приз, та ‘Again’, якщо він зможе зіграти ще раз.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1221101001001100114112015 | WinLoseAgainWinWin |

**Problem E-Черга**

**Черга**

|  |  |
| --- | --- |
| **Time limit:** | 500 ms |
| **Memory limit:** | 8 M |

Перед відкриттям кас, де продаються білети на залізничні потяги, вишикувалася величезна черга. Після відкриття кас виявилося, що працюють не всі вікна і час обслуговування обмежений. Самі покупці вирішили розподілитися по працюючим касам таким чином: не порушуючи послідовності людей у попередньо зайнятій черзі, до першого вікна переходить перша група покупців, до другого – наступна і т.д. Розподіл черги на групи повинен враховувати час роботи кас.

Вхідні дані:
у першому рядку міститься кількість людей у черзі ***N (1 ≤ N ≤ 3\*106)***, кількість працюючих вікон ***K (1 ≤ K ≤ 109)***, час роботи кас ***t***. У другому – час обслуговування ***a1, a2, …, an (1 ≤ ai ≤ t ≤109)*** кожної людини у черзі.

Вихідні дані:
одне число, яке визначає максимальний час, необхідний для обслуговування покупців у межах часу роботи кас ***t***.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 3 4 2 3 2 1 4 | 3 |
| 5 3 21 1 1 1 1 | 2 |
| 5 3 51 1 1 1 1 | 5 |