

# A. Sznurki

---

Dostępna pamięć: 64 MB

Jaś bawi się sznurkami. Sznurki mają długości będące liczbami naturalnymi. Jaś znajduje dwa najdłuższe sznurki tej samej długości i wiąże je ze sobą końcami, dostając w efekcie jeden sznurek o długości dwa razy dłuższej. Jaś postępuje w ten sposób tak długo, aż wszystkie sznurki będą miały taką różną długość. Jasia interesuje, ile sznurków będzie miał na końcu.

Przykładowo jeśli Jaś zaczyna z jednym sznurkiem długości 7, pięcioma sznurkami długości 6 i trzema sznurkami długości 3, to (multi)zbiory reprezentujące długości sznurków w kolejnych etapach zabawy wyglądają następująco:

$$\{3, 3, 3, 6, 6, 6, 6, 6, 7\} \rightarrow \{3, 3, 3, 6, 6, 6, 7, 12\} \quad (1)$$

$$\rightarrow \{3, 3, 3, 6, 7, 12, 12\} \quad (2)$$

$$\rightarrow \{3, 3, 3, 6, 7, 24\} \quad (3)$$

$$\rightarrow \{3, 6, 6, 7, 24\} \quad (4)$$

$$\rightarrow \{3, 7, 12, 24\} \quad (5)$$

Zatem na końcu (po pięciu wiązaniach sznurków) Jaś będzie miał 4 sznurki różnej długości.

**Uwaga:** W tym zadaniu zabronione jest używanie tych konstrukcji STL-a, których nazwy zawierają: `set`, `map` lub `queue`. Niedozwolone jest wykorzystanie gotowych odpowiedników tych konstrukcji w innych językach programowania.

## Specyfikacja danych wejściowych

W pierwszym wierszu danych wejściowych znajduje się liczba naturalna  $1 \leq m \leq 10^6$ , będąca liczbą różnych długości sznurka posiadanych początkowo przez Jasia. W każdym z kolejnych  $m$  wierszy znajduje się para dodatnich liczb naturalnych  $d, n_d$  oddzielonych spacją, oznaczająca, że Jaś ma  $n_d$  sznurków długości  $d$ , gdzie  $1 \leq d, n_d \leq 10^9$ . Dowolne dwie pary w danych wejściowych mają różne wartości  $d$ .

## Specyfikacja danych wyjściowych

W pierwszym i jedynym wierszu wyjścia Twój program powinien wypisać jedną liczbę naturalną będącą liczbą sznurków, które Jaś będzie miał na końcu zabawy.

### Przykład A

**Wejście:**

3  
3 3  
7 1  
6 5

**Wyjście:**

4

### Przykład B

**Wejście:**

1  
1 15

**Wyjście:**

4

### Przykład C

**Wejście:**

3  
1 4  
2 4  
3 4

**Wyjście:**

3