

## POLIBITS

### ANOS BISSEXTOS

Há muito tempo, as pessoas descobriram a tortura que é produzir um calendário. Um ano dura aproximadamente 365 dias, mas com uma pequena sobra de 5 horas e 48 minutos. Caso esta parte seja desprezada, a cada quatro anos, temos um adicional de cerca de 23 horas e 15 minutos. Imagine se, a cada ano, o calendário estivesse deslocado de um dia. Pode parecer pouca coisa, mas, ao longo de décadas, o calendário estaria completamente desfigurado; haverá tempos em que o verão ocorrerá no inverno; primavera em outono e assim por diante.

Percebendo o problema, o ditador Júlio César pediu ao astrônomo Sosígenes um modo de mitigar a discrepância. Baseando-se no que os egípcios faziam — um calendário de 365 dias, com o adicional de um dia a cada quatro anos — o estudioso então desenvolveu o **calendário Juliano**, com 365 dias divididos em 12 meses, mas com um dia adicional a cada quatro anos. Os anos em que esse dia é adicionado são denominados de **anos bissextos**, por conta da repetição do número 6 em 366 dias. Atualmente, os anos bissextos sempre caem nos anos múltiplos de 4. então, 1984, 2000, 1208, 2016 e 2024, por exemplo, são bissextos, pois são todos múltiplos de 4.

**Tarefa:** dados dois anos  $a$  e  $b$ , calcular a quantidade de anos bissextos de  $a$  até  $b$

**Restrições:**  $0 \leq a \leq b \leq 2 \cdot 10^9$

**Entrada:** dois inteiros  $a$  e  $b$

**Saída:** um inteiro  $n$ , sendo  $n$  a quantidade de anos bissextos no intervalo  $[a, b]$  ( $a$  e  $b$  inclusos).

<b>Exemplo de Entrada 1</b> 1996 2016	<b>Exemplo de Saída 1</b> 6
<b>Exemplo de Entrada 2</b> 1815 1997	<b>Exemplo de Saída 2</b> 46
<b>Exemplo de Entrada 3</b> 1405 1407	<b>Exemplo de Saída 3</b> 0

## Solução

---

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main(){
6      int a, b;
7      int quantidade, primeiro, ultimo;
8
9      // leitura dos anos
10     cin >> a >> b;
11
12     // calculo do primeiro ano bissexto
13     if(a % 4 == 0)
14         primeiro = a;
15     else
16         primeiro = a + 4 - (a % 4);
17
18     // cálculo do último ano bissexto
19     ultimo = b - (b % 4);
20
21     // cálculo e exibição
22     quantidade = 1 + (ultimo - primeiro)/4;
23     cout << quantidade << "\n";
24     return 0;
25 }
```

---

Podemos calcular a quantidade de bissextos facilmente se soubermos o primeiro e o último da lista. No exemplo 2, 1816 é o primeiro bissexto do intervalo e 1996 é o último. Em seguida, podemos contar a quantidade de números por meio de uma associação simples. Escrevendo explicitamente os anos bissextos, temos o seguinte:

(1816, 1820, 1824,  $\dots$ , 1988, 1992, 1996)

Dividindo por 4:

(454, 455, 456,  $\dots$ , 497, 498, 499)

Neste conjunto, temos  $499 - 454 + 1 = 46$  elementos, logo há 46 anos bissextos.

De forma mais geral, se  $x$  é o primeiro ano bissexto e  $y$  o último, então a quantidade de bissextos de  $x$  até  $y$ , com  $x$  e  $y$  inclusos, é dada por:

$$n = 1 + \frac{y - x}{4}$$

Podemos calcular  $x$  e  $y$ , basta somar ou subtrair os restos por 4, caso já não sejam divisíveis por 4. Por exemplo, 1815 deixa resto 3; se adicionarmos  $4 - 3 = 1$  a 1815, temos 1816, que

é nosso primeiro bissexto.

1997 deixa resto 1; se subtraímos 1 de 1997, temos 1996, que é nosso último bissexto. Em resumo:

Primeiro bissexto:

$$x = a + (4 - (a\%4))$$

Último bissexto:

$$y = b - (b\%4)$$

Quantidade total:

$$n = 1 + \frac{y - x}{4}$$

Devemos tomar o cuidado de analisar para o caso em que o intervalo é muito pequeno, por exemplo, de 1405 até 1407. Nesta situação, a fórmula ainda funciona. Então, temos que esta solução é geral e vale para todos os casos.