Problema A

Ver para crer

Problema

A sequência de inteiros S_n dita de *Schröder* modela o seguinte fenómeno combinatório: Numa grelha de tamanho $n \times n$, quantas formas tenho de caminhar do canto inferior esquerdo para o canto superior direito sem nunca transpor a diagonal $(0,0) - (n,n)$ e usando exclusivamente os caminhos para cima ($\uparrow$), para a direita ($\rightarrow$) e em diagonal montante à direita ($\nearrow$).

Por exemplo, para $S_2 = 6$, as soluções são (retirado do site wikipedia):

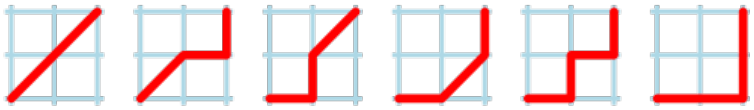

Figura 1: S_2 numa imagem

Para calcular S_n para um determinado n inteiro podemos nos socorrer de duas definições recursivas equivalentes:

$$\begin{array}{lcl} S_0 & = & 1 \\ S_1 & = & 2 \\ S_n & = & 3S_{n-1} + \Sigma_{k=1}^{n-2} S_k S_{n-k-1} \quad (com\ n > 1) \end{array}$$

ou

$$\begin{array}{lcl} S_0 & = & 1 \\ S_1 & = & 2 \\ S_n & = & \frac{6n-3}{n+1} S_{n-1} - \frac{n-2}{n+1} S_{n-2} \quad (com\ n > 1) \end{array}$$

Parece difícil acreditar que calculam a mesma sequência!

Nem sequer parecem ter a mesma complexidade. Por complexidade entendemos o número de chamadas a S. Assim, chamar S_0 ou S_1 nas duas definições tem o custo de 1. No entanto, chamar S_3 com a primeira definição não parece ter o mesmo custo que com a segunda definição!

A surpresa não acaba aqui!

Diz-se também que a sequência é diabólica, que a sua complexidade é monumental e que calcula valores rapidamente enormes.

A nós, é uma meia verdade! Afirmamos que se pode calcular a sequência em tempo linear mesmo se, de facto, os valores calculados tendem para o *enorme*.

Neste exercício, propomos validar ou invalidar experimentalmente estas afirmações.

Aviso:

1. Se esperamos uma implementação direta, fiel às definições da sequência para a primeira parte do exercício, esperamos uma implementação otimizada para a segunda parte do exercício.
2. A segunda parte do exercício precisa de uma cuidada implementação da sequência. Primeiro, é necessário uma implementação claramente eficiente, como referido no ponto anterior. Segundo, por esta considerar valores de entrada maiores, a sequência vai muito rapidamente devolver valores que ultrapassam claramente em tamanho o a capacidade dos inteiros. Para isso, aconselha-se o uso da aritmética de precisão arbitrária. Tal funcionalidade pode ser encontrada na biblioteca `zarith`.

Entrada

Uma linha com dois inteiros a e b separados por um espaço.

Saída

Uma primeira linha com dois inteiros u e v separados por um espaço. O inteiro u é o resultado de S_a pela primeira definição da sequência de *Schröder*. O inteiro v é o número de vezes em que a função recursiva que implementa *fielmente* a primeira definição de S é chamada.

Uma segunda linha com dois inteiros x e y separados por um espaço. O inteiro x é o resultado de S_a pela segunda definição da sequência de *Schröder*. O inteiro y é o número de vezes em que a função recursiva que implementa *fielmente* a segunda definição de S é chamada.

Uma terceira linha com um inteiro z. Este inteiro é S_b.

Limites

$0 \leq a \leq 20$ e $0 \leq b \leq 10000$

Exemplo de Entrada

```
6 100
```

Exemplo de Saída

```
1806 70
1806 25
2874761115350486026653425000745888138831358356111744362989662030744034089O
```

Explicação

Para o primeiro inteiro do input, 6, foi calculado o valor de S_6 com a primeira definição da sequência. O resultado foi 1806 e foi preciso executar 70 chamadas à função.

Na segunda linha, apresenta-se os mesmos resultados, mas usando a segunda definição da sequência. O que dá naturalmetne o mesmo valor para o termo S_6, 1806. Mas o número de chamadas à função é bem mais interessante, aqui 25.

A terceira linha apresenta o valor de S_{100}.