Crack the RSA code

The RSA code (named after the inventors of the algorithm: Rivest, Shamir and Adleman) essentially can't be cracked, because the integer factorization of large numbers can't be done efficiently. However, in actual practice, mistakes in the protocol or fragments of leaked information can tear the protection to pieces.

Given is a public key *s* of an RSA code, for example:

```
89998197990593928653354360537957796764
46475605769993162342218070514244053591
97872695613679745342773 03
```

Your task is to find the secret key, in other words two integer factors of the number s. Without any extra information, this is a mathematical job for a super computer. However, through espionage, the difference between both integer factors has leaked. For the example above, the difference v of the integer factors of s equals:

```
26847788780987879677717147747607682222
33499207622
```

Do not be afraid: to find the secret key you will only need secondary school algebra and some practical insight in numbers. However, you must correctly add and multiply numbers that are far too large for a calculator. Luckily, Python has no difficulty calculating with large integers without losing accuracy.

Assignment

Write a function crackRSA which can be used to crack an RSA code of which the key s and the difference v of two integer factors p_1 and p_2 of the key s are given. The key s and the difference v should be passed to the function as arguments. As a result, the functions should return the tuple $(p1, p2)$ with both integer factors of s, to which applies that $p_1 \leq p_2$ en $p_2 - p_1 = v$.

Example

```
>>> crackRSA(221, 4)
(13, 17)
>>> crackRSA(2183, 22)
(37, 59)
```

Het RSA-geheimschrift (vernoemd naar de uitvinders van het algoritme: Rivest, Shamir en Adleman) is in principe onkraakbaar, omdat het in twee priemfactoren ontbinden van een zeer groot getal niet efficiënt te doen is. In de praktijk kunnen fouten in het protocol of snippertjes uitgelekte informatie de beveiliging echter om zeep helpen.

Gegeven is een publieke sleutel *s* van een RSA-geheimschrift, bijvoorbeeld:

```
89998197990593928653354360537957796764
46475605769993162342218070514244053591
```

```
9787269561367974534277303
```

Jouw taak bestaat erin de geheime sleutel te vinden, ofwel de twee priemfactoren van het getal s. Zonder extra informatie is dit een rekenklus voor een supercomputer. Door spionage is het verschil tussen beide priemfactoren echter uitgelekt. Voor bovenstaand voorbeeld werd achterhaald dat het verschil v tussen beide priemfactoren van s gelijk is aan:

```
268477887809878796777171477476076822233499207622
```

Laat je niet afschrikken: om de geheime sleutel te vinden is nu niet meer dan algebra van de middelbare school en enig praktisch getalinzicht nodig. Je moet wel gehele getallen die veel te groot zijn voor een rekenmachientje exact optellen en vermenigvuldigen. Gelukkig kan Python zonder problemen rekenen met grote gehele getallen, zonder daarbij aan nauwkeurigheid in te boeten.

Opgave

Schrijf een functie `kraakRSA` waarmee een RSA code kan gekraakt worden waarvan de sleutel s en het verschil v tussen de twee priemfactoren p_1 en p_2 van de sleutel s gegeven zijn. De sleutel s en het verschil v moeten als argumenten aan de functie doorgegeven worden. Als resultaat moet de functie het tuple $(p1, p2)$ teruggeven met de twee priemfactoren van s, waarvoor geldt dat $p_1 \leq p_2$ en $p_2 - p_1 = v$.

Voorbeeld

```
>>> kraakRSA(221, 4)
(13, 17)
>>> kraakRSA(2183, 22)
(37, 59)
```