

Babylonian method

Some square roots like $\sqrt{25}$ or $\sqrt{81}$ are known by heart, but how can we compute for example $\sqrt{17}$? Nowadays, most people use a calculator or a computer to come up with the square root of a number s , but how was \sqrt{s} computed at times when calculators or computers weren't even invented?

Heron's method is perhaps the first algorithm used for approximating \sqrt{s} . It was named after the first-century Greek mathematician [Hero of Alexandria](#) who gave the first explicit description more than 2000 years ago of a method for computing the square root of any given number. This method is also known as the **Babylonian method**, named after the Babylonians who left us the oldest known [mathematical texts](#).



Babylonian clay tablet [YBC 7289](#). The diagonal displays an approximation of $\sqrt{2}$ in four sexagesimal figures: 1 24 51 10, which is good to about six decimal digits: $1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3} = 1.41421296\dots$. The tablet also contains an approximation of $\sqrt{1800}$ in three sexagesimal figures: 42 25 35, which equals $42.4263888\dots$.

The Babylonian method works as follows. Suppose we want to compute the square root of a strictly positive number $s \in \mathbb{R}$. We start with an initial estimate for \sqrt{s} , which we refer to as x_0 . Then, in each successive step we compute a more accurate approximation of the square root using the following formula: $x_{n+1} = \frac{1}{2} \left(x_n + \frac{s}{x_n} \right)$. Let's apply this algorithm for example to compute $\sqrt{17}$. We choose $x_0 = 6$ as an initial estimate of $\sqrt{17}$. In a next step we get the following more accurate approximation of the square root: $x_1 = \frac{1}{2} \left(6 + \frac{17}{6} \right) = 4.416666666666667$. Then, we repeat the same procedure to get an even more accurate approximation: $x_2 = \frac{1}{2} \left(x_1 + \frac{17}{x_1} \right) = 4.1328616352201255$. Along the same lines, we keep enhancing the accuracy of the approximation: $x_3 = 4.12311714060797$, $x_4 = 4.12310562563374$, $x_5 = 4.123105625617661$, $x_6 = 4.123105625617661$. The Babylonian method thus produces a series of numbers that converges to $\sqrt{17}$. We verify this result using `math.sqrt` from the `math` module:

```
>>> import math
>>> math.sqrt(17)
4.123105625617661
```

An approximation x_k resulting from step k is considered accurate enough when $\frac{|x_k - x_{k-1}|}{x_k} \leq 10^{-15}$. In that case it is no longer relevant to come up with

even more accurate approximations and the procedure stops.

Input

The first line of the input contains a number $s \in \mathbb{R}$. The second line of the input contains an initial estimate $x_0 \in \mathbb{R}$ of \sqrt{s} .

Output

The Babylonian method only works if both s and x_0 are strict positive numbers. If this is not the case, the message `invalid input` must be written to output. In case the input contains valid numbers, the Babylonian method must be applied to compute successive approximations of \sqrt{s} . For each step of the method, the index of x_i ($i = 0, 1, 2, \dots$) must be written to output, followed by a colon, a space and the approximation x_i . The method stops at the first step k where it holds that $\frac{|x_k - x_{k-1}|}{x_k} \leq 10^{-15}$ with x_k the approximation that results from step k and x_{k-1} the approximation that results from step $k-1$.

Example

Input:

```
17
6
```

Output:

```
0: 6.0
1: 4.416666666666667
2: 4.1328616352201255
3: 4.12311714060797
4: 4.12310562563374
5: 4.123105625617661
6: 4.123105625617661
```

Example

Input:

```
2
0
```

Output:

```
invalid input
```

Sommige vierkantswortels zoals $\sqrt{25}$ of $\sqrt{81}$ kennen we van buiten, maar hoe berekenen we bijvoorbeeld $\sqrt{17}$? Vandaag de dag wordt doorgaans een rekenmachine of een computer gebruikt om de vierkantswortel van een getal s te berekenen, maar hoe berekende men \sqrt{s} in de tijd dat er nog geen rekenmachines of computers bestonden?

De **methode van Heron** is wellicht het oudste algoritme voor de berekening van \sqrt{s} . Ze werd vernoemd naar de Griekse wiskundige [Heron van Alexandrië](#) die ruim 2000 jaar geleden

de eerste expliciete beschrijving gaf van een methode voor het berekenen van de vierkantswortel van een willekeurig getal. Deze methode is ook gekend als de **Babylonische methode** — een verwijzing naar de Babyloniërs die ons de oudste gekende [wiskundeteksten](#) achterlieten.



Babylonische kleitablet [YBC 7289](#). De diagonaal beschrijft een benadering van $\sqrt{2}$ met vier cijfers uit het 60-talig stelsel: 1 24 51 10. Dit komt ongeveer overeen met een benadering tot op zes decimale cijfers: $1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3} = 1.41421296\dots$. De tablet bevat voorts ook nog een benadering van $\sqrt{1800}$ met drie 60-tallige cijfers 42 25 35, wat gelijk is aan $42.4263888\dots$.

De Babylonische methode werkt op de volgende manier. Stel dat we de vierkantswortel willen berekenen van een strikt positief getal $s \in \mathbb{R}$. We starten met een initiële schatting van \sqrt{s} , en noemen die x_0 . Daarna berekenen we in elke stap een meer nauwkeurige benadering van de vierkantswortel op basis van de volgende formule: $x_{n+1} = \frac{1}{2} \left(x_n + \frac{s}{x_n} \right)$. Laten we dit algoritme bijvoorbeeld eens toepassen voor de berekening van $\sqrt{17}$. We kiezen $x_0 = 6$ als een initiële schatting voor $\sqrt{17}$. In een volgende stap bekomen we de volgende meer nauwkeurige benadering van de vierkantswortel: $x_1 = \frac{1}{2} \left(6 + \frac{17}{6} \right) = 4.416666666666667$. Daarna herhalen we de procedure om een nog nauwkeuriger benadering te bekomen: $x_2 = \frac{1}{2} \left(x_1 + \frac{17}{x_1} \right) = 4.1328616352201255$. Op dezelfde manier blijven we de nauwkeurigheid van de benadering verder verbeteren: $x_3 = 4.12311714060797$, $x_4 = 4.12310562563374$, $x_5 = 4.123105625617661$, $x_6 = 4.123105625617661$. De Babylonische methode resulteert dus in een rij van getallen die zal convergeren naar $\sqrt{17}$. We controleren dit resultaat met `math.sqrt` uit de `math` module:

```
>>> import math
>>> math.sqrt(17)
4.123105625617661
```

We beschouwen de benadering x_k uit stap k als voldoende nauwkeurig als $\frac{|x_k - x_{k-1}|}{x_k} \leq 10^{-15}$. Het is dan niet langer zinvol om een betere benadering te berekenen en we mogen de procedure daar stoppen.

Invoer

De eerste regel van de invoer bevat een getal $s \in \mathbb{R}$. De tweede regel van de invoer

bevat een initiële benadering $x_0 \in \mathbb{R}$ van \sqrt{s} .

Uitvoer

De Babylonische methode werkt alleen maar indien zowel s als x_0 strikt positieve getallen zijn. Indien dit niet het geval is, dan moet de boodschap ongeldige invoer uitgeschreven worden. Als de invoer wel geldig is, dan moet de Babylonische methode toegepast worden om opeenvolgende benadering van \sqrt{s} te bepalen. Voor elke stap moet het de index van x_i ($i = 0, 1, 2, \dots$) uitgeschreven worden, gevolgd door een dubbelpunt, een spatie en de benadering x_i . De methode stopt bij de eerste stap k waarvoor geldt dat $\frac{|x_k - x_{k-1}|}{x_k} \leq 10^{-15}$ met x_k de benadering die bekomen wordt in stap k en x_{k-1} de benadering die bekomen werd in de stap $k-1$.

Voorbeeld

Invoer:

17
6

Uitvoer:

0: 6.0
1: 4.416666666666667
2: 4.1328616352201255
3: 4.12311714060797
4: 4.12310562563374
5: 4.123105625617661
6: 4.123105625617661

Voorbeeld

Invoer:

2
0

Uitvoer:

ongeldige invoer