You have an unlimited number of coins of (n + 1) types, the cheapest of which has denomination 1 and each of the next types has denomination ai times greater than the previous one. You need to pay the sum *s* using as few coins as possible. Of course you can use multiple coins of the same denomination, but the sum of all coins must be equal to *s*.

**Input**

The first line of the file coins.in contains two integers separated by a space - **n** and **s** - the number of coins’ types, excluding the cheapest one, and the sum to pay.

The second line contains *n* integers separated by spaces - ai - the number of times each of the next coins is more expensive than the previous one.

**Output**

In the output file coins.out print a single integer – the minimum number of coins required to pay the sum s.

**Constraints**

1 ≤ *n* ≤ 105

0 ≤ s ≤ 109

2 ≤ ai ≤ 109

**Time limit: 1.0 sec**

**Memory limit: 256 MB**

**Example test**

|  |  |
| --- | --- |
| **Input (coins.in)** | **Output (coins.out)** |
| 3 42  3 2 2 | 4 |
| 3 2  3 4 5 | 2 |