Problem A
Seat Allocation

Consider an election where n parties run for m seats in parliament. Let v_i be the number of votes cast for party $i \in \{1, \dots, n\}$. To allocate the m seats to the parties, roughly proportional to their share of votes, one can use a system called *D'Hondt's method*. It works like this: A quotient is calculated for each party, initially $v_i/1$, the number of votes. The party with the largest quotient wins one seat, and its quotient is recalculated. This is repeated until all m seats is filled. The formula for the quotient is

$$\frac{v_i}{s_i + 1}$$

where s_i is the number of seats allocated to party i so far, initially 0 for all parties.

Input

On the first line of input: the number n of parties and m of seats, with $n, m \in \{1, \dots, 20\,000\}$, separated by a single space. On each of the following n lines, the integer v_i, satisfying $1 \leq v_i \leq 50\,000\,000$ and $v_1 + \cdots + v_n \geq m$. We also promise that the inputs are constructed so that the final seat can be uniquely determined—no tie breaking will be necessary.

Test groups

There are two tests groups. The inputs in test group 1, worth 80 of the 100 total points, satisfy the additional constraint $m \cdot v_i \leq 2^{31}$. This may avoid some rounding issues, depending on your choice of programming language and quotient calculations.

Output

The seat distribution, one per line, in the format shown in the example. The order is the same as in the input.

Explanation of Sample Input 4

Party 1 gets the first seat because it has the most votes; its quotient is recalculated to $\frac{17}{2}$. Party 2 gets the next seat because $10 > \frac{17}{2}$; its quotient is recalculated to $\frac{10}{2} = 5$. The third seat goes to Party 1, because $\frac{17}{2} > 5$, and the new quotient is $\frac{17}{3}$. Even the final seat goes to Party 1, because $\frac{17}{3} > 5$. In total, Party 1 gets 3 seats and Party 2 gets 1 seat.

Sample Input 1

```
2 2
10
10000000
```

Sample Output 1

```
0
2
```

Sample Input 2

```
2 3
12
11
```

Sample Output 2

```
2
1
```

Sample Input 3

```
2 4
12
11
```

Sample Output 3

```
2
2
```

Sample Input 4

```
2 4
17
10
```

Sample Output 4

```
3
1
```

Sample Input 5

```
4 14
38
35
36
37
```

Sample Output 5

```
4
3
3
4
```

CPU Time limit	3 seconds
Memory limit	1024 MB
Languages	Dansk, English
Author	Thore Husfeldt
Source	ITU BADS
License	