# Task: BUR
# Cat Burglars



The neighbourhood cats are a sneaky bunch. They walk around patios, sneak into houses, rummage through squirrel hideaways, and gather anything that looks tasty. In the past, cats got into petty squabbles over the loot and territory, but now Dexter is the boss and ensures that rules are followed.

Every week, all cats gather with all their loot. The treats are arranged into equivalent bundles, Dexter takes his share, and then oversees the split. Cats are in a strict hierarchy, numbered $1$ through $n$. They are all greedy, although some are greedier than the others, not necessarily in relation to their position in the hierarchy. The cats are also cunning and always make optimal decisions – it's the only way to survive in this cat-eat-cat world.

Assume there are $m$ equivalent bundles of treats. The cat with the lowest number in the hierarchy among those still **eligible** proposes a split, i.e. for each **eligible** cat $i$ proposes they receive $b_i$ bundles. This number can be zero, but cannot be negative, and the sum of all has to equal $m$. Then, all cats (including the one proposing the split) vote for or against the split. If at least half of the cats (rounded up) vote for, then the process is over, the loot is split, everyone shakes paws and returns to their various feline shenanigans. Otherwise, the split is deemed unfair and the proposing cat becomes **ineligible**, curls their tail up, and leaves the meeting in shame; the next cat in the hierarchy now proposes the split.

Because the cats act greedily and optimally, the way they propose the splits is consistent and predictable. Cat $i$ proposes a split based on the following strategy:

- If they know the situation is hopeless and there is no way their split gets accepted, they propose to take all the bundles by themselves and are rightly cast away.
- If there exists any split that will be accepted, they propose that. It is always better to receive $0$ bundles than to be cast away in shame.
- Among the acceptable splits they maximise the number of bundles they get (naturally).
- Cats know that it's prudent to suck up to those high in the hierarchy. Therefore, if there are multiple splits that maximise their share, they choose to assign more bundles to the cats high in the hierarchy. Specifically, they choose the split that *minimises* the number of bundles assigned to cat $i + 1$, then to cat $i + 2$, etc.

Cat's voting patterns are also consistent and predictable. The cat $i$ votes *for* the split, iff:

- it saves their skin, i.e. if the process advanced to their proposal they'd be cast away;
- $b_i \geq d_i + a_i$, where $d_i$ is the number of bundles they'd receive if they rejected the current proposal and $a_i$ is their *greed coefficient*.

All cats know each other's greed coefficients, always follow the same voting and proposal strategies, and know that others always act optimally as well. Dexter wants to automate this process, and so he asked you to write a program that given the number of bundles and all greed coefficients decides which cat receives how many bundles according to the above rules.

## Input

The first line of input contains two integers, $n$ and $m$, respectively the number of cats and the number of bundles to split.

In the second line there are $n$ integers $a_1, a_2, ..., a_n$ – the greed coefficients of all cats.

## Output

Your program should write one line to the standard output containing $n$ integers $b_1, b_2, ..., b_n$. If cat $i$ gets cast away in shame, $b_i = -1$; otherwise, $b_i$ is the number of bundles they receive.

## Examples

For the input:

```
3 100
28 1 56
```

the correct output is:

```
44 0 56
```

While for the input:

```
5 1
1 1 1 1 1
```

the correct output is:

```
-1 0 0 1 0
```

While for the input:

```
6 6
3 5 1 4 2 6
```

the correct output is:

```
2 0 0 0 4 0
```

**Explanation:** In the first example we have three cats, Apex, Binky, and Charlie. If Apex got cast away then Binky would propose a split where she receives all the $100$ bundles and Charlie gets nothing. Charlie would not be happy, but the vote would split 50/50 and Binky would have her way. Because of that, Apex has no way to convince Binky to vote for his split – he'd have to offer her at least $100 + 1$ bundles. Instead, he needs to convince Charlie by offering him enough bundles to satisfy his greed coefficient – at least $0 + 56$. He proposes $44, 0, 56$, and he and Charlie vote for the split.

In the second example we have Amy, Booper, Ciri, Dexter, and Eve. Amy is screwed – she doesn't have enough bundles to convince other cats, so she proposes to take the bundle for herself and gets cast away. Booper can choose one of two acceptable splits – give the bundle to Ciri or Dexter. According to the rules they choose the latter. Why are both splits acceptable? Assume both Booper and Ciri get thrown away. Then Dexter proposes to take all the bundles and wins. Therefore, $1$ bundle is enough for Ciri to convince Eve to vote for their split $(0, 0, 1)$. However, that means that both Ciri and Dexter can be convinced with just one bundle by Booper.

In the third example cats number $1$, $2$, and $5$ vote for the split proposed by cat number $1$. Cat number $2$ has no acceptable splits, so he prefers $0$ bundles to being cast away, while number $3$ can pass through a split that gives number $5$ two bundles.

### Additional examples

The following initial tests are also available:

- `0d` – $n = 10\,000$, $m = 1\,234\,567$, $a_i = \left\lfloor \frac{n-i}{200} \right\rfloor + 1$;
- `0e` – $n = 50\,000$, $m = 5\,000\,000$, $a_i = 64$ for all $i$.

## Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

In all tests $1 \le a_i \le 64$ for all $a_i$ and $1 \le m \le 5\,000\,000$.

| Subtask | Limits | Points |
| --- | --- | --- |
| 1. | $1 \le n \le 10\,000$ | $2$ |
| 2. | $1 \le n \le 50\,000$ | $8$ |