Finding Routers (routers)

There is a street of length l meters stretching from left to right, with n small routers occupying various distinct positions along it. The **origin** is defined to be the leftmost point of the street. The routers are labelled 0 to $n-1$ from left to right, and router i is placed $p[i]$ meters away from the origin.

It is guaranteed that router 0 is at the origin, and the distance in meters from each router to the origin is an **even integer**.

You wish to find out the position of each of the n routers. As the routers are very small and difficult to spot from afar, you've decided to use the following procedure to find them:

- Place a detector on a spot that is x meters away from the origin,
- Use the detector to find the label of the router closest to it. If there are two routers that are the same distance away from it, it will respond with the router with the smaller label.

You are allowed to use the detector at most q times. Devise a strategy to find the positions of all the routers.

Implementation Details

You should implement the following procedure:

```
int[] find_routers(int l, int n, int q)
```

- l: length of the street in meters.
- n: number of routers.
- q: maximum number of times the detector can be used.
- This procedure will be called exactly once by the grader.
- It should return an array p indicating the positions of each router, with $p[i]$ being the distance between router x and the origin.

The above procedure can make calls to the following procedure:

```
int use_detector(int x)
```

- x: distance between the detector and the origin.
- x must be at least 0 and at most l.
- This procedure will return the label of the router closest to the detector. If there are two routers

that are the same distance away from it, it will return the smaller label.
- This procedure may be called no more than q times.

Examples

Example 1

Consider the following call:

```
find_routers(5, 2, 10)
```

There are 2 routers on a street of length 5 meters and you are allowed at most 10 calls to `use_detector`. Suppose the routers are placed at 0 and 4 meters away from the origin respectively.

The `find_routers` procedure may choose to call `use_detector(3)`, which returns 1, as router 1 at position 4 is closest to the detector.

The `find_routers` procedure may then choose to call `use_detector(2)`, which returns 0, as both routers 0 and 1 are the same distance away from the detector and router 0 has a smaller label.

At this point, there is sufficient information to conclude that the routers are at positions 0 and 4 respectively.

As such, the `find_routers` procedure should return [0, 4].

Example 2

Consider the following call:

```
find_routers(6, 3, 10)
```

There are 3 routers on a street of length 6 meters and you are allowed at most 10 calls to `use_detector`. Suppose the routers are placed at 0, 2 and 6 meters away from the origin respectively.

The `find_routers` procedure may choose to call `use_detector(5)`, which returns 2 as router 2 is at position 6 and therefore is the closest to the detector.

The `find_routers` procedure may then choose to call `use_detector(4)`, which returns 1, as routers 1 and 2 are an equal distance away from the detector.

At this point, there is sufficient information to conclude that the routers are at positions 0, 2 and 6 respectively.

As such, the `find_routers` procedure should return [$0,\ 2,\ 6$].

Constraints

- $p[0] = 0$
- $0 \le p[i] \le l$ and $p[i]$ is even. (for all $0 \le i \le n-1$)
- $p[i] < p[i+1]$ (for all $0 \le i \le n-2$)
- $5 \le l \le 100\ 000$

Subtasks

1. (16 points) $l = 100\ 000,\ n = 2,\ q = 100\ 001$
2. (21 points) $l = 100\ 000,\ n = 100,\ q = 100\ 001$
3. (23 points) $l = 100\ 000,\ n = 2,\ q = 20$
4. (40 points) $l = 100\ 000,\ n = 1000,\ q = 20\ 000$

In addition, Subtask 4 will be a partial scoring subtask. Let m be the maximum number of times `use_detector` is called among all testcases.

- If $m > 20000$, you will score 0 points.
- If $7500 < m \le 20\ 000$, you will score $\frac{20\ 000 - m}{12\ 500} \cdot 40$ points.
- If $m \le 7500$, you will score 40 points.

Sample grader

The sample grader reads the input in the following format:

- line 1: $l\ \ n\ \ q$
- line 2: $p[0]\ \ p[1]\ \ \ldots\ \ p[n-1]$

The sample grader prints your answers in the following format:

- line 1: $p[0]\ \ p[1]\ \ \ldots\ \ p[n-1]$ as reported by `find_routers`.
- line 2: the number of calls to `use_detector`.