Task 3: Solar Storm (`solarstorm`)

Squeaky the Mouse is the captain of a spaceship, on a mission to explore the outer reaches of the solar system. His spaceship is shaped like a rod – there is a long, straight pathway along the length of the spaceship, and N modules are connected at various locations along the pathway. The modules are numbered from 1 to N, starting from the left side of the spaceship. The distances between adjacent modules need not be equal.

Here is an example of a possible configuration of the spaceship:

Each module i (where i ranges from 1 to N) contains equipment to support the operation of the spaceship, and hence has an associated positive integer value v_i that measures how important it is for the operation of the spaceship. Electrical wiring is run between every pair of adjacent modules, which allows the equipment to be controlled remotely from another module.

Alas, there is a impending solar storm quickly approaching Squeaky's spaceship! The solar storm contains a burst of electromagnetically charged particles which will damage the equipment on Squeaky's spaceship if they are not protected properly.

Luckily, Squeaky's competent crew has brought along S shields on this mission. Each shield may be deployed in any module, or left undeployed. When deployed, each shield generates a magnetic field that will deflect the electromagnetically charged particles away from the spaceship, and hence protect all equipment located in modules no more than K metres away from the shield. (Equipment located in the same module as a shield will of course be protected.)

Note that the pathway does not need to be shielded as the electrical wiring along the pathway is not susceptible to damage from the charged particles.

To ensure that as much of the spaceship remains operational after the solar storm as possible, Squeaky wants to deploy the shields optimally, i.e. to maximise the total value of protected modules. Furthermore, from any protected module, he wants it to be possible to control equipment in all protected modules, to minimise the amount of walking that the crew will need to do after the solar storm. However, due to the design of the electrical wiring on the spaceship, a module A can control equipment at some module B only if all modules between A and B (inclusive of both A and B) are operational. As Squeaky's chief engineering officer on this mission, help him to figure out the optimal placement of shields satisfying this constraint.

Input

Your program must read from standard input.

The first line contains three integers, N, S, and K, which represent the number of modules, number of shields, and the protection radius of each shield (in metres) respectively.

The second line contains $N - 1$ integers. The i^{th} integer is d_i, the distance between module i and module $i + 1$, in metres.

The third line contains N integers. The i^{th} integer is v_i, the value of module i.

Output

Your program must print to standard output.

The output should contain exactly two lines.

The first line should contain a single integer, T, the number of shields to deploy.

The second line should contain exactly T integers, specifying the modules in which shields should be deployed. This list may be printed in any order. If there are multiple possible ways to maximise the total value of protected modules, all of them will be accepted.

Implementation Note

As the input lengths for subtasks 2, 3, 4, 6, and 7 may be very large, you are recommended to use C++ with fast input routines to solve this problem. The scientific committee does not have a solution written in Java or Python that can fully solve this problem.

C++ and Java source files containing fast input/output templates have been provided in the attachment. You are strongly recommended to use these templates.

If you are implementing your solution in Java, please name your file `SolarStorm.java` and place your `main` function inside `class SolarStorm`.

Subtasks

The maximum execution time on each instance is 2.0s, and the maximum memory usage on each instance is 256MiB. For all testcases, the input will satisfy the following bounds:

- $1 \le S \le N \le 10^6$
- $1 \le K \le 10^{12}$
- $1 \le d_i \le 10^6$ for all $1 \le i \le N - 1$
- $1 \le v_i \le 10^6$ for all $1 \le i \le N$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	10	$S = 1$, $N \leq 10^4$, $K \leq 10^9$, $d_i \leq 10^5$ for all $1 \leq i \leq N - 1$, $v_i \leq 10^5$ for all $1 \leq i \leq N$
2	7	$S = 1$, $d_i = 1$ for all $1 \leq i \leq N - 1$
3	11	$S = 1$
4	8	$K = 1$, $d_i = 2$ for all $1 \leq i \leq N - 1$
5	18	$N \leq 10^4$
6	16	$S \leq 50$
7	30	-

Sample Testcase 1

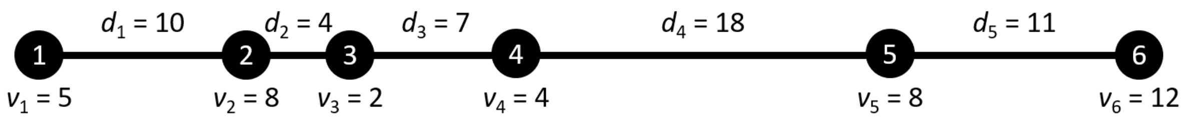
This testcase is valid for subtasks 5, 6, and 7.

Input	Output

`6 2 7`
`10 4 7 18 11`
`5 8 2 4 8 12`
	`2`
	`3 5`

Sample Testcase 1 Explanation

The spaceship looks like this:



The optimal locations to deploy the two shields are at modules 3 and 5. The shield at module 3 will protect modules 2, 3, and 4; the shield at module 5 will protect module 5. Notice that the pathway between modules 4 and 5 is not fully protected; this is acceptable as the electrical wiring along the pathway is not susceptible to damage.

Note that deploying the two shields at modules 3 and 6 instead is not a valid solution, because module 5 will be damaged by the solar storm, making it impossible for crew at module 6 to control equipment at modules 2, 3, or 4.

Sample Testcase 2

This testcase is valid for subtasks 5, 6, and 7.

Input	Output

`6 2 38`
`10 4 7 18 11`
`5 8 2 4 8 12`
	`1`
	`4`

Sample Testcase 2 Explanation

The spaceship has the same configuration as in sample testcase 1, but the shield radius is much larger. Deploying a single shield at module 4 is sufficient to protect all modules.

Note that there are many other acceptable solutions. Some alternative solutions could be:

- Deploy a single shield at module 3
- Deploy two shields, one at module 3 and the other at module 5
- Deploy two shields at module 3

All valid solutions that maximise the total value of protected modules will be accepted.

Sample Testcase 3

This testcase is valid for subtasks 1, 3, 5, 6, and 7.

Input	Output

`6 1 12`
`10 4 7 18 11`
`5 8 2 4 8 12`
	`1`
	`5`

Sample Testcase 3 Explanation

The spaceship has the same configuration as in sample testcase 1. Deploying the shield at module 5 will protect modules 5 and 6, which is optimal.

Note that it is also optimal to deploy the shield at module 6.

Sample Testcase 4

This testcase is valid for subtasks 1, 2, 3, 5, 6, and 7.

Input	Output
12 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 3 4 5 6 6 5 4 3 2 1	1 6

Sample Testcase 4 Explanation

The optimal location is to deploy the shield at module 6, which will protect modules 4, 5, 6, 7, and 8.

Note that it is also optimal to deploy the shield at module 7.

Sample Testcase 5

This testcase is valid for subtasks 4, 5, 6, and 7.

Input	Output
10 3 1 2 2 2 2 2 2 2 2 2 3 7 5 6 8 4 3 2 2 9	3 3 4 5

Sample Testcase 5 Explanation

The optimal locations to deploys the three shields are at modules 3, 4, and 5.