HPC³ 2024
Problem F, English
Pax Galactica
Maximum Points: 50

You are an intergalactic trader travelling from solar system to solar system. A solar system is a rectangular prism in a 3-dimensional cartesian coordinate system with one of its corners at (0, 0, 0) and width W, height H, and length L ($1 \leq W \leq 10^3,\ 1 \leq H \leq 10^3,\ 1 \leq L \leq 10^3$). You enter the solar system at a coordinate within the rectangle ($x_A,\ y_A,\ z_A$) and must move to the exit point at another coordinate within the rectangle ($x_E,\ y_E,\ z_E$). Your ship is fitted with a warp drive to traverse the vast distances of space. The drive allows you to teleport from your current point to any given point within the rectangle. However, there are a number of planets given by an array of coordinates B of length $3l$ ($0 \leq l \leq 50$) where each ($B_{i_x},\ B_{i_y},\ B_{i_z}$) is the position of a planet. The gravitational field of each planet interferes with the drive. Each field is a perfect sphere with a centre at the location of each planet and the radius of each field is given by an integer array R ($1 \leq R_i \leq 100$) of length l. If you use the drive to teleport through the gravitational field of the planet, you will be trapped in the void forever. Formally, if you draw the shortest line between the points that you warp between and there exists a point on that line such that the distance between that point and any planet is less than the radius of the field of that planet, you fail.

You want to get through the system quickly, so find an array of coordinate points P of smallest arbitrary size v within the rectangle such that starting at the entry point, warping sequentially to each point in P, then warping to the exit point does not result in you crossing the gravitational fields of any planet.

Subproblem 1

The problem as described, find a set of points P such that lines drawn sequentially between each point do not intersect spheres B of radii R.

Given $W,\ H,\ L,\ l,\ x_A,\ y_A,\ z_A, x_E,\ y_E,\ z_E,\ B,$ and R, return P.

Input format

The first line of each input contains 10 integers $l,\ W,\ H,\ L,\ x_A,\ y_A,\ z_A,\ x_E,\ y_E,$ and z_E.
The second line of each input contains $3l$ integers: The content of array B.
The third line of each input contains l integers: The content of array R.

```
l  W  H  L  xA  yA  zA  xE  yE  zE
B[0][0]  B[0][1]  B[0][2]  …  B[l-1][0]  B[l-1][1]  B[l-1][2]
R[0]  R[1]  R[2]  …  R[l-1]
```

Output format

The first line of each input contains 1 integer v.
The second line of each output contains $3v$ integers: The content of array P.

```
v
P[0][0]  P[0][1]  P[0][2]  …  P[v-1][0]  P[v-1][1] P[v-1][2]
```

Example Test Cases

Input 1

```
3  14  26  50  4  14  7  48  14  7
15  13  7  36  16  7  46  18  7
7  6  3
```

Output 1

```
1
7  2  7
```

A line between the entry and (7, 2, 7) does not cross any fields, a line between (7, 2, 7) and the exit does not cross any fields, and 1 is the smallest length of P. So, the program could output this. Note that there are many valid possible Ps.

Subproblem 2

You have just purchased a big upgrade to your ship that will allow you to traverse solar systems with greater accuracy, a fine-point drive. It operates exactly the same as your old drive but can process real number points. The problem is still the same as described, find a set of points P such that lines drawn sequentially between each point do not intersect spheres B of radii R. However, all values can be real numbers instead of integers.

Given $W,\ H,\ L,\ l,\ x_A,\ y_A,\ z_A, x_E,\ y_E,\ z_E,\ B,$ and R, return P.

Notes

- Because answers must be reasonably gradable, answers will be rounded to 5 decimal places. So, this problem can be solved using numbers with 5 decimal places.

Input format

The first line of each input contains 1 integer l and 9 real values $W,\ H,\ L,\ x_A,\ y_A,\ z_A,\ x_E,\ y_E,$ and z_E.
The second line of each input contains $3l$ real values: The content of array B.
The third line of each input contains l real values: The content of array R.

```
l  W  H  L  xA  yA  zA  xE  yE  zE
B[0][0]  B[0][1]  B[0][2]  …  B[l-1][0]  B[l-1][1]  B[l-1][2]
R[0]  R[1]  R[2]  …  R[l-1]
```

Output format

The first line of each input contains 1 integer v.
The second line of each output contains $3v$ real values: The content of array P.

```
v
P[0][0]  P[0][1]  P[0][2]  …  P[v-1][0]  P[v-1][1] P[v-1][2]
```

Example Test Cases

Input 1

```
4  20.5  30.5  40.5  3.85  9.75  12.25  18.35  25.15  30.65
5.3  15.3  35.4  15.6  15.8  30.3  14.98  16.7  20.8  5.25  15.8  30.88
4.75  6.25  5.09  5.555
```

Output 1

```
2
10.25  5.25  20.25  10.25  25.25  20.25
```

A line between the entry and (10.25, 5.25, 20.25) does not cross any fields, a line between (10.25, 5.25, 20.25) and (10.25, 25.25, 20.25) does not cross any fields, a line between (10.25, 25.25, 20.25) and the exit does not cross any fields, and 2 is the smallest length of P. So, the program could output this.