South African Computer Olympiad
Final Round 2010
Day 1

Overview

Problem	missiles	bentopia	seating	funroll
Source	missiles.java missiles.py missiles.c missiles.cpp	bentopia.java bentopia.py bentopia.c bentopia.cpp	N/A	funroll.java funroll.py funroll.c funroll.cpp
Input file	stdin	stdin	seating.in	stdin
Output file	stdout	stdout	seating.out	stdout
Time limit	0.2 seconds	0.7 seconds	N/A	1 second
Memory limit	128MiB	128MiB	N/A	128MiB
Number of tests	10	10	10	10
Points per test	10	10	10	10
Detailed feedback	Yes	No	No	No
Total points	**100**	**100**	**100**	**100**

The maximum total score is 400 points.

(Don't) Launch the Missiles

Introduction

The Nuclear Missile is the most dangerous ride in the theme park. To ensure that it can only be run by authorized personnel, it requires a long numeric code to operate.

To make sure they didn't forget the code, the ride operators wrote it down on a piece of paper. To ensure that nobody else could read it, they encrypted it using a simple cipher.

Unfortunately, they have forgotten the cipher key, and only remember a small portion of the security code.

Task

Write a program to determine the cipher key.

The security code consists of a sequence of numbers; the cipher works by adding the non-zero numeric key, K, to each number in the code.

The ride operators remember that the sequence `23 87 56 77` appeared somewhere in the security code.

If there is more than one possible key, find any one of them.

Example

If the encrypted code is `28 34 98 67 88 91 23`, then the key must be 11 since this would mean the code is `17 23 87 56 77 80 12`. This code contains the remembered sequence, while a different key would give a code which would not.

Input (stdin)

The first line of input contains an integer, N. The second line contains N space-separated integers: the encrypted code.

Sample input

```
7
28 34 98 67 88 91 23
```

Output (stdout)

Output a single integer, K.

Sample output

```
11
```

Constraints

- $4 \leq N \leq 10\,000$
- $-10\,000 \leq$ each number in the unencrypted code $\leq 10\,000$
- $-10\,000 \leq K \leq 10\,000$

Additionally, in test runs worth 50%, either

- $-10 \leq K \leq 10$ or
- $N \leq 100$

Time limit

0.2 seconds. Python multiplier: 10. Java multiplier: 2.

Detailed feedback

Detailed feedback is enabled for this problem.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Bentopian Scare House

Introduction

The haunted house of Bentopia is known to be the scariest in the universe. Bruce and Carl have decided to visit Bentopia to visit the haunted house for themselves. To enhance their experience, there is a strict rule that only one person is allowed in the haunted house at a time.

Carl will enter the haunted house first, followed tentatively by Bruce. While Carl is going for the thrill, Bruce is not very fond of haunted houses. He would like to spend as little time in each room of the house as possible, and he certainly won't visit any rooms or go through any doors unless Carl tells him they are safe.

Task

Carl has just returned from the haunted house. He has told Bruce his exact path through the house. Carl's path contains exactly N rooms. Each room in the path is labelled by an arbitrary integer, A_i. A_1 is the starting room and A_N is the ending room. Two consecutive integers A_i and A_{i+1} in the path means that you can get from room A_i to room A_{i+1} via a one-way door.

Given Carl's path, find a path from start to end which visits each room at most once.

Example

Carl has told Bruce that his path is: $1, 2, 3, 4, 1, 3$. The starting room is 1 and the ending room is 3. There are two paths between these two rooms which visits no room more than once: $1, 2, 3$ and $1, 3$

Input (stdin)

The first line of input consists of a single integer, N, the number of rooms in Carl's path. Each of the following N lines contains a single integer, A_i, the ith room in Carl's path.

Sample input

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

Output (stdout)

On the first line, output the length of Bruce's path. On each subsequent line, output a single room number of Bruce's path in order. In the case of multiple solutions, any one will be accepted.

Sample output

```
2
1
3
```

Constraints

- $1 \leq N \leq 50\,000$
- $1 \leq A_i \leq 50\,000$

Additionally, in 60% of the test cases:

- $1 \leq N \leq 5\,000$

Additionally, in 30% of the test cases:

- $1 \leq N \leq 20$

Time limit

0.7 seconds. Python multiplier: 10. Java multiplier: 2.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Seating Arrangement

Introduction

Young Æthelred and his friends are finally old enough to go on the legendary Helicokinesis — a fearsome thrill ride which is rumoured to offer a different experience depending on where one chooses to sit. Æthelred takes fun very seriously and is determined for him and his friends to enjoy as wide a range of dizzying actions as possible. Æthelred only has time for two rides before the park closes, but luckily no one else is riding on the Helicokinesis at this time. Æthelred has thus decided to let everyone choose where they will sit for the first ride from the ride's $N \times M$ rectangular grid of seats. However, for their second ride, he wishes to assign his friends to seats such that everyone is far away from the positions they sat in before.

Task

Given the initial positions of the K children, help Æthelred determine an arrangement that allows him and his friends to all sit far away from their initial positions. More specifically, if each child initially sits at position (r_i, c_i) and then sits at position (r'_i, c'_i) for the second ride, then the score of that arrangement is defined to be the sum of $\sqrt[4]{(r_i - r'_i)^2 + (c_i - c'_i)^2}$ (the square root of the distance) calculated for each child. Find an arrangement with as high a score as possible.

Example

The seats are arranged in a 3x3 grid and there are 5 children, numbered 1 to 5. Initially, the children chose to sit in the first two rows as illustrated. A good seating arrangement is obtained by the first row moving to the back and the second row moving forward.

1	2	3
4	5	.
.	.	.

Initial configuration

5	.	4
.	.	.
3	2	1

Rearrangement

In this configuration, the first and third children are displaced $\sqrt{8}$ units, the second 2 units, the fourth $\sqrt{5}$ and the fifth $\sqrt{2}$. This yields a score of approximately 7.46.

Input (seating.in)

The first line of the input contains three space-separated integers, N, M and K. The next K lines each contain two space-separated integers, r_i and c_i, the row and column where the ith child is initially sitting.

Sample input

```
3 3 5
1 1
1 2
1 3
2 1
2 2
```

Output (seating.out)

Output K lines, the ith of which contains two space-separated integers, the row and column where the ith child should sit on the second ride.

Sample output

```
3 3
3 2
3 1
1 3
1 1
```

Constraints

- $N \cdot M \leq 200\,000$
- $1 \leq K \leq N \cdot M$
- $1 \leq r_i \leq N$ for each child
- $1 \leq c_i \leq M$ for each child

Submission

Each submitted output file will be checked to ensure that it is formatted correctly, and that every child is placed in a valid seat.

Scoring

Invalid submissions will score 0.

Valid submissions for each case will be compared to the best submission for that case by any contestant or member of the scientific commitee. If your submission is the best you will score 10.

Otherwise, if the best arrangement has a score of B and your arrangement has a score of A, then your submission will score $\lfloor 10 \cdot e^{25(\frac{A}{B}-1)} \rfloor$ where e is the base of the natural logarithm, approximately 2.71828.

Funroller

Introduction

The Eight-Planets theme park is designing a new type of rollercoaster: The funroller. The key to the funroller is that there are no horizontal sections of track. In fact, all funrollers can be made out of tracks sloping 45 degrees upwards and tracks sloping 45 degrees downwards.

The Eight-Planets council have approached you to help them.

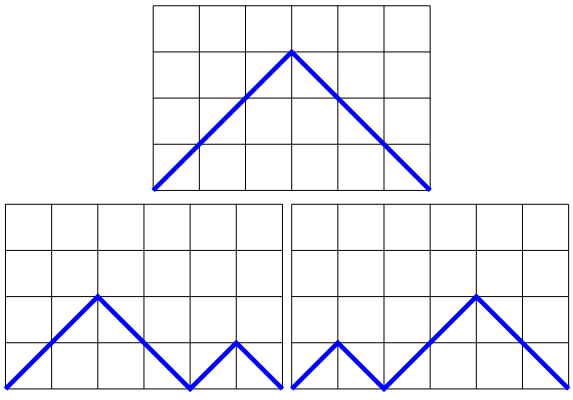
Task

The council have asked you to calculate how many different funrollers can be made with exactly N up tracks and N down tracks.

Funrollers always start and end at ground level. For health and safety reasons, the height of a funroller cannot fall below the ground at any point. To ensure that the ride is not predictable, a funroller must not have left-right symmetry. In other words, riding it backwards must be different to riding it forwards.

The inhabitants of the Eight-Planets theme park are known to have only 4 993 fingers, so the council would like you to tell them the remainder when the output is divided by 4 993.

Example



One invalid (above) and two valid funrollers

As can be seen above, there are only 2 funrollers that can be made using exactly 3 pairs of tracks. The first funroller is made of two up tracks followed by two down tracks, another up track and finally a down track. The other is the reverse of this.

Input (stdin)

The first and only line of input contains N, the number of pairs of up and down tracks.

Sample input

3

Output (stdout)

Output a single integer, the remainder after dividing the number of different funrollers by 4 993.

Sample output

2

Constraints

- $1 \leq N \leq 1\,000$

Additionally, in 30% of the test cases:

- $1 \leq N \leq 10$

Time limit

1 second. Python multiplier: 10. Java multiplier: 2.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.