## Enumeration

## Task ID: enum Time Limit: 1 second

## Problem

Write a program to read an integer $N$ and output $N^{2}$ integers in the format specified below.

## Input

The input contains an integer $N(1 \leq N \leq 10)$.

## Output

The output consists of $N$ lines, each containing $N$ integers, separated by a single space. The $N$ integers in the first row are 1 to $N$ respectively. The $j$-th integer on the $i$-th row is equal to the minimum value of $j$ and $N+1-i$.

## Sample tests

| Input | Output |  |  |  |
| :--- | :--- | :--- | :--- | :--- |
| 4 | 1 | 2 | 3 | 4 |
|  | 1 | 2 | 3 | 3 |
|  | 1 | 2 | 2 | 2 |
|  | 1 | 1 | 1 | 1 |

HKOI 2012 Senior Group

# Infinite Coding 

## Task ID: infc $\quad$ Time Limit: 1 second

## Problem

As you may know, some people like coding very much. A friend of Dr. Jones, Joe is one of them. Let's take a look at his daily timetable:

| Time | Activity | Cost |
| :---: | :---: | :---: |
| $0000-0300$ | Playing computer games | 1 |
| $0300-0600$ | Sleeping | 3 |
| $0600-0900$ | Coding | 1 |
| $0900-1200$ | Studying | 2 |
| $1200-1500$ | Coding | 5 |
| $1500-1800$ | Doing homework | 0 |
| $1800-2100$ | Coding | 1 |
| $2100-0000$ | Sleeping | 3 |

He divides each day into 8 session and 3 of them are used for coding. At the beginning, he felt quite satisfied with the schedule. However, sometimes he writes long programs that take days to complete it. In this case, he will prefer coding for 9 hours continuously instead of coding for 3 hours 3 times a day. Therefore, he wants to reschedule his timetable to maximize the time of continuous coding.

Joe achieves this by the following method: select an activity in a time slot, then reschedule to somewhere else. Of course, some activities are important that he doesn't want to move and some are less important. Therefore, there is a cost for moving each activity. In the case above, he may move the first Coding session after the Study time with cost 1. Then, move doing homework to the end of each day with no cost. Here is his new schedule:

| Time | Activity | Cost |
| :---: | :---: | :---: |
| $0000-0300$ | Playing computer games | 1 |
| $0300-0600$ | Sleeping | 3 |
| $0600-0900$ | Studying | 2 |
| $0900-1200$ | Coding | 1 |
| $1200-1500$ | Coding | 5 |
| $1500-1800$ | Coding | 1 |
| $1800-2100$ | Sleeping | 3 |
| $2100-0000$ | Doing homework | 0 |

Note that when he reschdeules some activity, it may affect many other activities. But the cost he counted is only the activity he 'moves'. Now he is satisfied with his new schedule because he can code continuously for 9 hours every day! Listening to his story, Joe's friends feel very admirable for Joe and they want to follow Joe. Since Joe has many friends, it is better to write a program to do it. And this is your task.

Here is the formal problem discription: the day is divided into $N$ sessions, $M$ of them are used for coding. Originally, activity $i$ is scheduled in the $i$-th time session. The cost for moving activity $i$ is $w_{i}$. Find the minimum cost such that the coding sessions are continuous.

Note that when we say continuous, we may cross over midnight. (See sample 2)

## Input

The first line contains integer $N, M$.
The second line contains $N$ integers $w_{1}, w_{2}, \ldots, w_{n}$.

HKOI 2012 Senior Group

The third line contains $M$ unique integers in increasing order, the indices of activity that is coding.

## Output

A single integer, the minimum cost.

## Sample tests

| Input | Output | Input | Output |
| :---: | :---: | :---: | :---: |
| 83 | 1 | 105 | 6 |
| 13125013 |  | 9923641999999 |  |
| 357 |  | 146910 |  |

## Constraints

In test cases worth $25 \%, N \leq 100$.
In test cases worth $50 \%, N \leq 3000$.
In all test cases, $1 \leq M \leq N \leq 500,000,0 \leq w_{i} \leq 1000$.

## Toothpicks

## Task ID: tooth Time Limit: 1 second

## Problem

Dr. Jones' son is always curious to everything he first touches. This time, they go to 'Yum Cha' together. As usual, many toothpicks are provided. And as usual, Dr. Jones' son is playing with the toothpicks again.

After a while, Dr. Jones suggests to play a game with his son. He draws some non-overlapping rectangles on a paper. The rectangles are always parallel to the side of the paper. After that, he asks his son to use toothpicks to reconstruct the rectangles as on the paper. Note that the pattern should be filled by $1 \times 1$ squares. Each toothpick has a length of 1 . This is an example using 13 toothpicks to construct a pattern made by two rectangles.


Dr. Jones want to estimate how many toothpicks is required before his son finish the pattern. You, sitting on the next table, hear this story and decide to find out what is the minimum number of toothpicks required.

## Input

The first line contains an integer $N$, the number of rectangles. It is guranteed that the rectangles will not overlap each other.
The following $N$ lines each contains 4 integers. The first two locate the lower-left point of rectangle $i$ and the last two locate the upper-right point. Each rectangle will have a non-zero area.

## Output

The number of toothpicks required.
Hints: The answer may exceed 32-bit integer.

## Sample tests

| Input |  | Output |  |  |
| :--- | :--- | :--- | :--- | :--- |
| 2 |  |  |  | 13 |
| 0 | 0 | 2 | 1 |  |
| 1 | 1 | 3 | 2 |  |


| Input |  |  | Output |  |
| :--- | :--- | :--- | :--- | :--- |
| 6 |  |  |  | 202 |
| 6 | 1 | 9 | 4 |  |
| 7 | 4 | 8 | 14 |  |
| 5 | 14 | 10 | 17 |  |
| 10 | 0 | 16 | 6 |  |
| 17 | 0 | 18 | 1 |  |
| 17 | 1 | 20 | 4 |  |

## Constraints

In test cases worth $50 \%, N \leq 100$, the absolute value of coordinates is less than 100 .
In all test cases, $N \leq 1000$, the absolute value of coordinates is less than $10^{7}$.

HKOI 2012 Senior Group

## Songs

| Task ID: song | Time Limit: 1 second |
| :--- | :--- |

## Problem

To your surprise, Dr. Jones is an excellent musician!
Recently he has composed a new song. The song has magical power that it can prevent diseases. By listening to that song, people can become invincible for a while. So kind is he, he wants to perform that song to the public so that people will not get ill anymore. Unfortunately, he only has one musical instrument with him, and that instrument can only play a certain number of musical notes. Dr. Jones knows what notes are contained in the song, and what notes that the instrument can play. He would like to know if he can play the magical song with the instrument.

When Dr. Jones composed the song, he had foreseen that it is very likely that the instrument cannot play the song in its original key. Dr. Jones therefore composed the song in a brilliant way that the healing power still persists when the song is played in another key. However, the healing power decreases when the key gets higher, so he wants to play the song at the lowest key possible.

In this question, musical notes are represented by positive integers. Increasing the key of the song by $x$ means that all notes of the song are increased by $x$. Note that $x$ should be non-negative. For example, the magical song may contain the notes $1,3,4$ and 5 , and the instrument may play $2,3,4,5$ and 6 . The instrument cannot play the song in the original key (key 0) because the note 1 cannot be played. On the other hand, the instrument can play the song in key 1 because the notes $2,4,5,6$ can all be played. Thus, key 1 is the lowest key possible that the instrument can play.

Since you are an excellent and helpful programmer, Dr. Jones would like you to write a program that can find out whether the instrument can play the song in some key. If it is possible, he would also like to know the lowest possible key that he can play in.

You will be given two lists of positive integers. The first list contains $A$ unique integers that are the notes contained in the song. The second list contains $B$ unique integers that are the notes that the instrument can play. Both lists are sorted in ascending order. To make your task easier, you will also be given the highest note $N$ in the song.

## Input

The first line contains 3 integers, $A, B$ and $N$.
The second line contains $A$ integers, the notes in the song.
The third line contains $B$ integers, the notes that the instrument can play.

## Output

If the song cannot be played in any key, output in one line 'Impossible' without quotes. If the song can be played, output in one line the lowest key possible.

## Sample tests

| Input | Output | Input | Output |
| :---: | :---: | :---: | :---: |
| 455 | 1 | 455 | Impossible |
| 1245 |  | 1245 |  |
| 23456 |  | 13579 |  |

## Constraints

In test cases worth $20 \%, 1 \leq A, B \leq 100$
In test cases worth $50 \%, 1 \leq A, B \leq 3000, A \leq N \leq 3000$
In all test cases, $1 \leq A, B \leq 3000, A \leq N \leq 1,000,000$.

HKOI 2012 Senior Group

# Gene Mutation ${ }^{1}$ 

| Task ID: gene | Time Limit: 1 second |
| :--- | :--- |

## Problem

A 'supervirus' is now in an outbreak in the city! One of the most threatening characteristic of this 'supervirus' is that the infected person is asymptomatic during the first week of infection. However, after the incubation period, he will have skin rash over the whole body and die within 5 hours due to exaggerated immune response.

Another threatening feature of 'supervirus' is that the virus is very infectious. It can be transmitted by direct contact of the saliva of the infected person, so if he talks to someone, the virus is easily spreaded to that person.

Fortunately, Dr. Jones has invented a drug that can be injected to the infected people to cure them. Since the drug targets at the deoxyribonucleic acid (DNA in short) molecule inside the virus, the efficacy of the drug will be reduced if the 'supervirus' is mutated.
Dr. Jones discover that the gene of a 'supervirus' can be represented by a cycle of $N$ distinct positive integers, and they are from 1 to $N$.
For example, when $N=5$, the gene of the 'supervirus' is shown in figure 1 .
Before mutation, exactly one of the $N$ bonds in the gene must be broken down first. Then, it will mutate for several times by swapping several pairs of adjacent numbers, and then form the broken bond again. For example, after 2 mutations, the gene may become (see figure 2 ):



Figure 1


Figure 2
Dr. Jones cannot predict which bond will be broken, but he has confirmed that each 'supervirus' can break down one bond and form that bond again for at most once.

[^0]The current drug can be slightly modified to treat different mutated viruses, but it will have no therapeutic effect anymore when the gene becomes a cycle of $1,2, \ldots, N$ in clockwise direction (see figure 3 ).


Figure 3
Researchers call this an 'ultravirus'.
In order to estimate the time left for inventing new drugs before the mutation of 'supervirus' into mutate into 'ultravirus', he asks you to find the minimum number of mutations needed.

## Input

The first line is the integer $N$.
The second line contains the number in the gene of the 'supervirus', arranged in clockwise direction.

## Output

There is only 1 line: the minimum number of mutations required for 'supervirus' to become 'ultravirus'.

## Sample tests

| Input |  | Output | Input |  | Output |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 5 |  |  |  |  | 2 |  |  |
| 1 | 2 | 4 | 5 | 3 |  |  |  |

## Explanation

In the first sample, first break down the bond between 1 and 3. After that, swap 5 and 3, and then 4 and 3 , and form the broken bond again. (see figure 4)
In the second sample, it is equivalent to the first sample because the gene is cyclic.

## Constraints

In test cases worth $30 \%, 1 \leq N \leq 8$.
In test cases worth $50 \%, 1 \leq N \leq 60$.
In test cases worth $70 \%, 1 \leq N \leq 300$.
In all test cases, $1 \leq N \leq 5000$.


Figure 4


[^0]:    1'Mutation' means 'sudden change'

