Question 1 - Profit Maximization

Emilia, a quantitative researcher, predicts how the closing share price of a stock moves over time.

She wants to find the maximum possible profit of a stock over a given period of time, using only **one** buy and **one** sell operation according to a given sequence of predicted share prices.

Constraints

- Short selling is not allowed.
- All of the predicted share prices are positive integers.

Input format

The first integer input is the number of predicted days.

The subsequent integer input is a sequence of positive integers. The element at position i refers to the predicted share price of a given stock on the ith day.

Output format

An integer which is the maximum possible profit with only **one** buy and **one** sell operation.

Examples

Example 1

Input

```
14 5 1 6 3 2 5 6 1 3 6 2 5 5 10
```

`Number of predicted days = 14`

`Sequence of predicted share prices = [5,1,6,3,2,5,6,1,3,6,2,5,5,10]`

Output

`9`

i.e. Buy the stock on day 1 for $1 and sell the stock on the last day at $10.

Example 2

Input

```
8 100 10 12 5 6 14 5 6
```

`Number of predicted days = 8`

`A sequence of predicted share prices = [100, 10, 12, 5, 6, 14, 5, 6]`

Output

`9`

i.e. Buy the stock on day 3 for $5 and sell the stock on day 5 at $14.

Question – 2 Profit Maximization - Extended

Emilia can now use multiple buy and sell operations over a given period of time.

Given a sequence of predicted share prices, she wants to find the maximum possible profit while using the smallest number of trading operations throughout the given time.

Constraints

- Short selling is not allowed.
- All of the predicted share prices are positive integers.
- You can only execute one buy or one sell operation of a share on a given day.
- Only one share can be bought or sold at a time.
- You are not required to execute a buy or sell operation every day.

Input format

The first integer input is the number of predicted days.

The subsequent integer input is a sequence of positive integers. The element at position i refers to the predicted share price of a given stock on the ith day.

Output format

An integer that is the maximum profit using the smallest number of trading operations throughout the given time.

Examples

Examples 1

Input

```
14 5 1 6 3 2 5 6 1 3 6 2 5 5 10
```

`Number of predicted days = 14`

`A sequence of predicted share prices = [5,1,6,3,2,5,6,1,3,6,2,5,5,10]`

Output

```
22
```

i.e. The smallest operation required is 8. Emilia can achieve this by buying at i = 1 when the price is $1 and selling at i = 2 when the price is $6 for a profit of $5. She can then buy at i = 4 when the price is $2 and sell at i = 6 when the price is $6 for a profit of $4. She can then buy at i = 7 when the price is $1 and sell at i = 9 when the price is $6 for a profit of $5. Then buy at i = 10 when the price is $2 and sell at i = 13 when the price is $10 for a profit of $8. If you add up all of the profits from these buy and sell orders (5 + 4 + 5 + 8) the output is 22.

Examples 2

Input

```
8 100 10 12 5 6 14 5 6
```

`Number of predicted days = 8`

`A sequence of predicted share prices = [100, 10, 12, 5, 6, 14, 5, 6]`

Output

```
12
```

i.e. The smallest operation required is 6. Emilia can achieve this by buying at i = 1 when the price is $10 and selling at i = 2 when the price is $12 for a profit of $2. Then she can buy at i = 3 when the price is $5 and sell at i = 5 when the price is $14 for a profit of $9. Then buy at i = 6 when the price is $5 and sell at i = 7 when the price is $6 for a profit of $1. If you add up all of the profits from these buy and sell orders (2 + 9 + 1) the output is 12.

Question – 3 Profit Model for John

John has recently started stock trading. He has predicted share prices for a particular company, over the next N days. John wants to analyze this data to build a model which will predict the best day to buy and sell the shares to achieve a specific profit. If there are multiple approaches of buying and selling shares to achieve this profit, John would like to know which of these will achieve the profit the earliest.

Input format

The first line contains two integers N and D, where N is the number of days for which he is predicting the share values and D is the number of different profits he would like to achieve.

The next line contains N space separated integers, where N_i is the value of the share on the i+1th day.

The next D lines contain a single integer D_i, where D_i is the profit that needs to be made.

Constraints

- Only 1 share can be bought.
- Short selling is not allowed.
- $1 \le N \le 100000$
- $1 \le D \le 10$
- $1 \le N_i, D_i \le 1000000$

Output format

Print in the same line two space separated integers - the day on which the share was bought and the day on which the share was sold. The buy and sell days for different profits should be separated by `,`. If it is not possible to achieve the desirable profit, print -1.

Examples

Example 1

Input

```
6 2

3 1 2 1 4 5

3

2
```

Output

`4 5,3 5`

i.e. To achieve a profit of 3, John can either buy on day 2 or day 4 and sell on day 5 or he can buy on day 3 and sell on day 6. The approach which takes the minimum number of days is where he buys on day 4 and sells on day 5. So, the answer is 4 5. To achieve a profit of 2, John can either buy on day 1 and sell on day 6 or he can buy on day 3 and sell on day 5. The approach which happens earliest is where John buys on day 3 and sells on day 5.

Example 2

Input

```
6 2

3 6 9 8 2 4

5

2
```

Output

```
1 4,2 4
```

i.e. To achieve a profit of 5, John can buy on day 1 when the price is 3 and sell on day 4 when the price is 8 for a profit of 5. To achieve a profit of 2, he has two options, buy on day 2 and sell on day 4 or buy on day 5 and sell on day 6. The approach which happens earliest is where John buys on day 2 and sells on day 4.

Question – 4 Risk Trading

Whenever a company trades securities, there are various risks involved with the trade. Risk analysis is done for each trade in order to make the maximum profit from that trade. Each available trade can have the following properties:

- Probability that the trade will make a profit (p).
- Probability that the trade will make a loss (1-p).
- Potential profit of the trade (x).
- Potential loss of the trade (y).

Find and print the maximum expected amount of money the company can make by performing at most m of the n trades, given the values of m, n, x, y and p.

Input format

The first line contains two space-separated integers denoting the respective values n (the number of trades available) and m (the maximum number of trades allowed).

The second line contains p space-separated floating-point numbers describing the respective values of p_i, where each p_i denotes the probability that the ith transaction will result in a profit.

The third line contains x space-separated floating-point numbers describing the respective values of x_i, where each x_i denotes the potential profit of the ith transaction.

The fourth line contains y space-separated floating-point numbers describing the respective values of y_i, where each y_i denotes the potential loss of the ith transaction.

Constraints

- $1 \le n, m \le 100000$
- $0 \le x, y \le 100$
- $0 \le p \le 1$
- All x, y and p are floating-point numbers scaled to exactly two decimal places (i.e 2.45 format).

Output format

Print the maximum expected amount of money that can be made by performing at most m of the n available trades. Scale your answer to exactly 2 decimal places.

Examples

Example 1

Input

```
4 2

0.50 0.50 0.50 0.50

4.00 1.00 2.00 3.00

4.00 0.50 1.00 1.00
```

Output

`1.50`

i.e. There are n = 4 transactions available and we can perform at most m = 2 of them. We also know the probability that each transaction results in a profit is 0.5. If the third and the fourth transactions are performed, the expected amount of money made from these transactions is: (0.5 * 2.0) - ((1 - 0.5) * 1.0) + (0.5 * 3.0) - ((1 - 0.5) * 1.0) = 1.5. Since this is greater than all of the other possibilities we could calculate, 1.50 is our answer.

Example 2

Input

```
2 2

0.90 0.50
```

```
1.00 0.50

100.00 0.40
```

Output

```
0.05
```

i.e. There are n = 2 transactions available and we can perform at most m = 2 of them. We know that the probability the first transaction is profitable is 0.9 and for the second transaction this is 0.5. If the second transaction is performed, the expected amount of money made from this transaction is: (0.5 * 0.5) - ((1 - 0.5) * 0.4) = 0.05. Since this is greater than all of the other possibilities we could calculate, we print 0.05 as our answer.

Question – 5 Perfect Matching

Credit Suisse organizes a private banking career information session for its potential future private bankers and currently employed private bankers. There are n private bankers and m participants.

Assume for each participant, they want to meet a number of private bankers and similarly, for each private banker they want to recruit a number of participants. However, only one-on-one meetings are possible. So for each session, one participant can only meet one banker.

If Credit Suisse has a list of preferences from participants and private bankers, how many sessions are needed in order to fulfil everyone's preferences?

Constraints

- Every banker and participant must have at least one preference.

Input format

The first line relates to the private bankers, and the second line relates to the participants.

The first integer in each line is the number of bankers/participants.

The subsequent integer input is the preference of bankers/participants, the preference of each person is separated by `,`.

For example:

```
2 1&2,2
2 1,2
```

The first line of input means that there are two private bankers. The preference of banker 1 is to meet participants 1 & 2, and the preference of banker 2 is to meet participant 2 only.

The second line of input means that there are two participants. The preference of participant 1 is to meet banker 1 only, and the preference of participant 2 is to meet banker 2 only.

Output format

An integer that is the minimum number of sessions required to fulfil everyone's preferences.

Examples

Example 1

Input

```
2 1,2&3
3 1,2,2
```

Output

`2`

i.e. In the first session, banker 1 will meet with participant 1 (fulfilling banker 1's preference and participant 1's preference) and banker 2 will meet with participant 2 (fulfilling banker 2's first preference and participant 2's preference). Another session is needed to fulfil banker 2's second preference and participant 3's only preference. In the second session, banker 2 will meet with participant 3 (fulfilling banker 2's second preference and banker 3's preference) and banker 1 will meet with participant 2. After two sessions, everyone's preferences have been fulfilled.

Example 2

Input

```
3 1,1,1
3 3,1,1
```

Output

3

i.e. There are many ways to arrive at this solution. This is just one of them. In the first session, banker 1 will meet with participant 1 (fulfilling banker 1's preference), banker 2 will meet with participant 2 and banker 3 will meet with participant 3. In the second session, banker 2 will meet with participant 1 (fulfilling banker 2's preference), banker 3 will meet with participant 2, and banker 1 will meet with participant 3 (fulfilling participant 3's preference). This leaves banker 3, participant 1, and participant 2's preferences. So in the third session, banker 3 will meet with participant 1 (fulfilling banker 3's preference and participant 1's preference), banker 1 will meet with participant 2 (fulfilling participant 2's preference), and banker 2 will meet with participant 3. After three sessions, everyone's preferences have been fulfilled.

Question – 6 Encrypting Messages

Data encryption prevents data visibility in the event of its unauthorized access.

Consider the following encryption algorithm to encipher a given string input. Firstly, discard all spaces of the string. Then store all the characters within a matrix, according to the constraints below, to get the encoded string output.

Constraints

- `floor(squareRoot(stringLength)) <= matrixRows <= matrixColumns <= ceil(squareRoot(stringLength))`
- `matrixRows x matrixColumns >= stringLength`
- Choose the matrix with the smallest area.
- Print out the characters of the first column, then embed a space before printing out the following column, etc.

Input format

A string

Output format

An encrypted string

Examples

Example 1

Command line input:

```
coding
```

Output: `ci on dg`

i.e. The string length is 6. The square root of 6 is between 2 and 3. Thus, the string is rewritten as a matrix with 2 rows and 3 columns.

```
cod
```

```
ing
```

Example 2

Command line input:

```
its harder to read code than to write it
```

Output: `ideeoi teatwt srdhr htcai aoont rrdte`

i.e. The string length is 32. The square root of 32 is between 5 and 6. However, 5 x 6 is not >= 32, therefore, the string is rewritten as a matrix with 6 rows and 6 columns.

```
itshar
dertor
eadcod
ethant
owrite
it
```

Question – 7 Q&A community

There is a Q&A community where users can raise questions and answer questions raised by others, to earn credits. However, there are some users within the community who are suspected to have cheated, therefore, they require validation.

The following cases are treated as suspicious:

- User A answers User B's question and User B answers User A's question
- If two or more users, which are considered suspicious, answer user C's question then user C is also considered suspicious.

Find the suspicious users.

Constraints

- Every user can only ask a maximum of one question.
- Not every user needs to ask or answer a question, at least 2 users do.
- A user can answer questions without asking a question.
- A user cannot answer their own question.
- 1 < userId < 10000
- 1 < Number of questions < 10000

Input format

The first line contains only one integer, which is the number of questions answered in the community.

The second line contains integers which refer to questions. The first integer is the questioner's ID and the subsequent integer(s) is the answerer's ID(s). Each question detail is separated by `,`.

For example:

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

There are three questions answered in the community. User 1's question is answered by User 2. User 2's question is answered by User 1. User 3's question is answered by Users 1 & 2.

Output format

A string that contains all suspicious User ID(s), sorted in ascending order, separated by `,`.

For example:

```
1,2,3
```

Examples

Example 1

Command line input:

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

i.e. Users 1 and 2 answer each other's questions, so they are both suspicious. Suspicious Users 1 and 2 answer User 3's question, so User 3 is suspicious.

Output: `1,2,3`

Example 2

Command line input:

```
4
1 2,2 1,3 1 4,4 1 2
```

i.e. Users 1 and 2 answer each other's questions, so they are both suspicious. Suspicious Users 1 and 2 answer User 4's question, so User 4 is suspicious. Suspicious Users 1 and 4 answer User 3's question, so User 3 is suspicious.

Output: `1,2,3,4`

Question – 8 Counting Change

James has recently started working in a currency exchange office at the airport. As many people are buying different types of currencies before they go on holiday, he ends up working with various types of coins. James would like to find all possible ways of making change for a desired amount using different coins.

For example, if James has 3 different types of coins, and the value of each is given as 15, 31 and 9, an amount of 63 can be made in two ways: {9, 9, 15, 15, 15} and {9, 9, 9, 9, 9, 9, 9}.

Create a countNumberOfWays function which returns an integer denoting the number of possible ways to give change.

Input format

The first line:

- n: an integer, is the desired amount.
- m: an integer, is the number of different coin types.

The second line:

- coins: space-separated distinct integers describing the respective values of each coin.

Output format

An integer denoting the number of possible ways to make change for the desired amount.

Examples

Example 1

Input

```
12 4
```

```
2 3 4 5
```

Output

```
10
```

i.e. There are 10 ways to change n = 12 using coins with values 2, 3, 4 and 5.

1. {2, 2, 2, 2, 2, 2}
2. {2, 2, 2, 2, 4}
3. {2, 2, 2, 3, 3}
4. {2, 2, 4, 4}
5. {2, 2, 5, 3}
6. {2, 3, 3, 4}
7. {3, 3, 3, 3}
8. {2, 5, 5}
9. {3, 4, 5}
10. {4, 4, 4}

Thus, we print 10 as our answer.

Example 2

Input

```
206 4

2 20 9 30
```

Output

```
214
```

Question – 9 Unauthorized Transactions

It is important that banks recognize unauthorized transactions in order to protect their clients. Leon has recently started working at a large credit card company where his role is to investigate fraudulent credit card transactions. He is attempting to sort transactions, depending on their fraud probability, into separate boxes.

Let t = 2 mean that Leon has 2 types of transactions and 2 different boxes, both labelled from 0 to t-1. The current organization of the transactions in each box can be shown using a matrix M (size t **x** t). Consider M = [[8, 3], [3, 9]]:

	Type 0	**Type 1**
Box 0	8	3
Box 1	3	9

In this table, we can see in box 0 there are 8 transactions of type 0 and 3 transactions of type 1. In box 1, there are 3 transactions of type 0 and 9 transactions of type 1. Leon is able to switch, in a single operation, two transactions in different boxes. He can switch a type 0 transaction from Box 1 with a type 1 transaction from Box 0. As shown below.

	Type 0	**Type 1**
Box 0	9	2

	Type 0	Type 1
Box 1	2	10

He can continue doing this until he has all transactions of type 0 in box 0 and all transactions of type 1 in box 1. The sorted boxes are reflected in the Matrix table below. There can be multiple different ways of sorting the transactions.

	Type 0	Type 1
Box 0	11	0
Box 1	0	12

These switching operations need to fulfil the following condition in order for the transactions to be sorted:

- Every box has only transactions of the same type. Two transactions of the same type cannot be located in two different boxes.

Input format

The first line contains an integer n, the number of unsorted problems. Attempt to sort n different unsorted problems, each in the form of a matrix M.

Each of the next n sets contains:

- Integer t represents the number of boxes (rows) and transaction types (columns).

- The next t lines contains integers, separated using a space, for row M[i].

Constraints

- A box is a two dimensional array of integers, illustrating the number of transactions of each type found in each box.
- 1 ≤ n ≤10
- 1 ≤ t ≤ 100.
- 0 ≤ M[box][Transaction type] ≤ 100000

Output format

For each unsorted problem, print the string “Possible” if Leon can sort the transactions in the given matrix. Else, print the string “Impossible”. These strings should be separated by `,` when answering each unsorted problem.

Examples

Example 1

Input

```
2

3

2 4 0

3 0 1

1 0 0

2

1 4
```

```
5 4
```

Output

`Possible,Impossible`

Explanation

We perform the following n = 2 unsorted problems.

The table below shows one way for which the first unsorted problem can be solved. Thus, we print “Possible”.

	Type 0	**Type 1**	**Type 2**
Box 0	6	0	0
Box 1	0	4	0
Box 2	0	0	1

The table below shows the matrix for the second unsorted problem:

	Type 0	**Type 1**
Box 0	1	4
Box 1	5	4

No matter how many times we attempt to switch transactions of type 0 and 1, we will never end up with box 0 only containing type 0 transactions and box 1 only containing type 1 transactions. Thus, we print "Impossible".