United Kingdom and Ireland Programming Contest 2018

20th October 2018

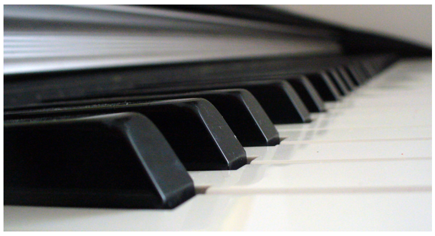
Problems

- A Analogue Cluster
- B Bus Logic
- C Code Word
- D Dynamo Wheel
- E Evenly Divided
- F Fibonacci Compression
- G Garden Variety Vampire
- H Hamster Ball
- I Internet Upload
- J Jackpot
- K Kings
- L Last Word

Problems are not ordered by difficulty.
Do not open before the contest has started.

This page is intentionally left (almost) blank.

Problem A
Analogue Cluster



For his thesis in distributed systems, Drew is focusing on an underserved market ripe for disruption: the player piano, a genus of analogue device which, given a piece of paper of appropriate style and width, can play whatever song is copied onto it.

Drew is going to network these devices; they will communicate using their native medium of long pieces of paper.

He hit a snag immediately: pianos can only communicate directly if they take the same width of paper, and not all of the pianos in the computer science department take the same width of paper. Some will need to be retrofitted if his plans are to succeed.

Time is valuable and, in particular, the time of the expensive technician Drew has hired to carry out the work is eye-wateringly valuable. What is the smallest number of pianos he can get away with modifying to make his project work?

Input

- One line containing the number of pianos, n ($1 \le n \le 1000$), followed by the number of connections between pianos, c ($1 \le c \le 10^5$).
- One line containing the n integer widths of each piano's paper intake in centimetres, in order, w_1 to w_n ($1 \le w \le 10^6$).
- Each of the following e lines are all distinct, and each contains two integers a and b ($1 \le a < b \le n$) indicating that pianos a and b need to become compatible.

Output

Output the minimum number of pianos that can be modified in order for all of the connections to become possible.

Sample Input 1	**Sample Output 1**

`4 2`
`40 20 40 2`
`2 3`
`1 3`
	`1`

Problem B
Bus Logic

A core part of the decision of where to live in a city like Nottingham is the availability of transport links to interesting places. This is particularly intersting to Max, who enlivens his stressful life as organiser of UKIEPC by making frequent sightseeing travels around town in a bright orange bus.

Max's idea of a good time is a visit to a spot that takes exactly one bus journey to get to. He is considering moving house to be near to one specific spot along his favourite bus route—how many such other scenic spots can he reach from there (assuming that on a given trip he can choose a new bus route each time)?

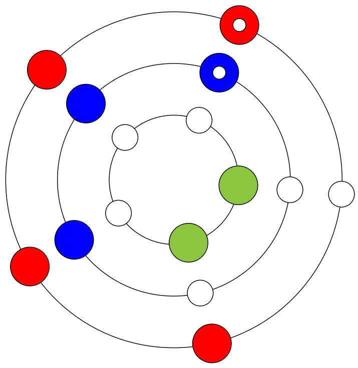


Figure B.1: A bus route map illustrating Sample Input 1. Max, as usual, is drawn as a white dot in the centre of each bus stop he can start from.

Input

- The first line of input contains three integers: Max's starting stop, m $(1 \leq m \leq s)$. the number of buses, b $(1 \leq b \leq 50)$, and the number of stops, s $(1 \leq s \leq 50)$.
- The next b lines contain the bus routes, each written as a string of s characters where having the ith character as '1' denotes that this bus route has a stop at i, and '0' denotes that it does not.

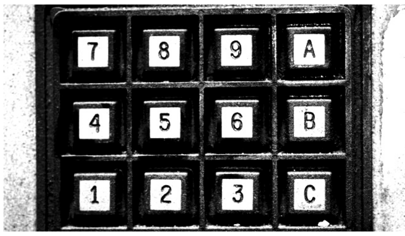
Output

Output the maximum number of other stops Max can reach from the starting stop by taking exactly one bus.

Sample Input 1	Sample Output 1
1 3 5 01100 10011 10111	3

Sample Input 2	Sample Output 2
2 2 3 101 101	0

Problem C
Code Word



Following on from the engineering success of *Left Pad, Ltd*, Lynn started a new venture: *Internet of Security, Inc*. The company's flagship product will be a device for entering passcodes.

The chief innovation of this device will be in its ability to reject attempts to set insecure passwords. An insecure passcode is defined as a sequence of digit presses such that at least two consecutive presses are either directly or diagonally adjacent.

Lynn, ever-vigilant for the future, is worried that this system might not allow enough unique passcodes to support a large company with trillions of employees. For a given digit pad grid size, and fixed length of password, calculate the number of allowed passwords.

Since the number might be very large in some cases, output your answer mod $1\,000\,000\,007$.

Input

- The first line contains the integers r and c ($1 \le r, c \le 100$), the number of rows and columns of buttons on the pad.
- The second line contains the integer l ($1 \le l \le 200$), the single allowed length for any passcode.

Output

Output the number of legal passcodes, mod $1\,000\,000\,007$.

Sample Input 1	Sample Output 1
3 3 2	32

Sample Input 2	Sample Output 2
100 1 5	860286658

Sample Input 3	Sample Output 3
49 97 191	814099263

Problem D
Dynamo Wheel

A water wheel is a device for converting the potential energy of water into the kinetic energy of rotation. We have attached buckets along our otherwise perfectly-balanced wheel. Each bucket has a specific weight when empty, and a specific weight when full of water, both in kilograms.

Whenever a bucket reaches the top of the water wheel, it is filled to its maximum capacity. When the bucket reaches the bottom, diametrically opposite, it is emptied again.

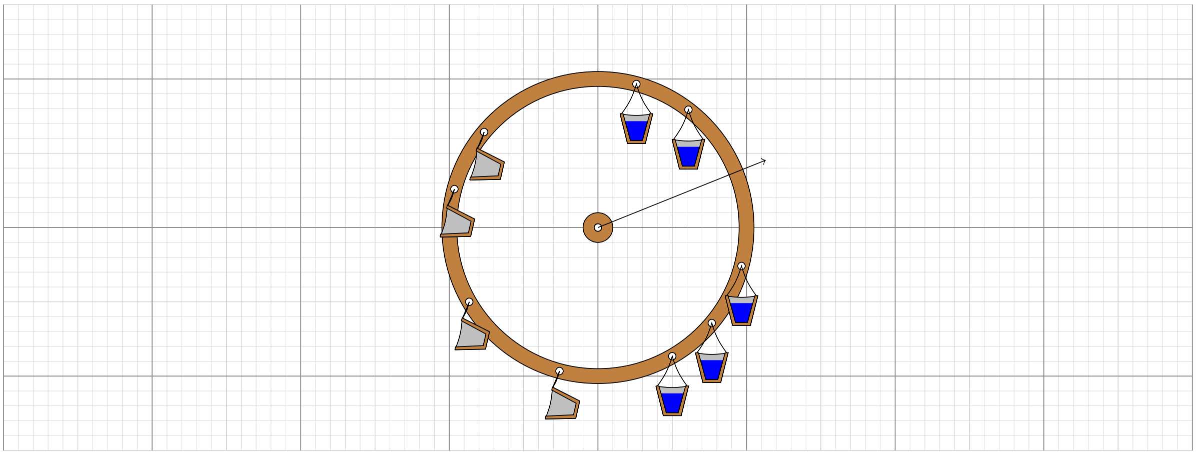


Figure D.1: Illustration of Sample Input 1 at approximately 15.0 degrees clockwise from the start. The centre of mass is drawn as a vector from the origin.

The acceleration of the wheel depends on the horizontal distance between the centre of mass of the wheel and its axis of rotation. Centre of mass of the wheel at some time is defined as the sum of the co-ordinates of the buckets at that time multiplied by their weight.

Our water wheel has a radius of exactly 1 metre. What is the maximum positive x-component of centre of mass achieved by the wheel at any angle?

Input

- The first line of input contains the number of buckets on the wheel, n ($2 \leq n \leq 10^5$).
- The remaining n lines of input each contain a description of a bucket, in angular order:
  - the decimal clockwise angle of the ith bucket in degrees, d_i ($0.0 \leq d \leq 360.0$),
  - the decimal weights of this bucket when empty and full, e_i and f_i ($0 \leq e_i \leq f_i \leq 10^4$).

Output

Output the maximum horizontal component of the centre of mass of the wheel over all possible angles. Your answer should be within an absolute or relative error of at most 10^{-6}.

Sample Input 1	Sample Output 1
9 0.000 0.5 1.8 22.50 1.0 1.7 90.00 0.5 1.0 115.0 1.0 1.2 135.0 1.0 1.2 180.0 1.0 1.2 225.0 1.0 1.2 270.0 1.0 1.2 295.0 1.0 1.2	1.65664252966349700991

Problem E
Evenly Divided

The Association of Chartered Mountaineers oversaw a sharp resurgence in membership this year, and must now face the inevitable strains of growth: the group photo they usually take can no longer fit everyone in one long row.

Members have been split into two groups: Tall and Short, so that the picture can be doubled up with taller people standing behind shorter people in two rows of $\frac{n}{2}$ each.

Every cloud has a silver lining, especially when mountaineering. This is an opportunity for the members to meet people. Many new joiners. were assigned a mentor from the members who had already signed up before they joined; the organisation wants to choose a way of arranging the rows such that nobody is standing directly in front of or behind their mentor, assuming they have one.

Find a way of arranging the two rows such that this is possible. The number of tall people is always the same as the number of short people.

Input

The input consists of:

- a line consisting of the number of members in the mountaineering society, which is a positive even integer m ($1 \leq m \leq 10^5$).
- m further lines, with the ith line ($1 \leq i \leq m$) consisting of an integer indicating whether the ith member is short (0) or tall (1), then the number of the ith member's mentor, t_i ($0 \leq t \leq m$). When $t_i = i$, this indicates that the ith member did not have a mentor.

Output

If an arrangement is possible, output 2 lines of $\frac{n}{2}$ numbers each to show which member should stand where.

Every number of type 1 should occur somewhere on the first row, and every number of type 0 should occur somewhere on the second row. Nobody should share a column with their mentor.

Otherwise, output `impossible`.

Sample Input 1

```
4
0 1
1 1
1 2
0 3
```

Sample Output 1

```
3 2
1 4
```

Sample Input 2

```
4
0 1
1 1
0 1
1 1
```

Sample Output 2

```
impossible
```

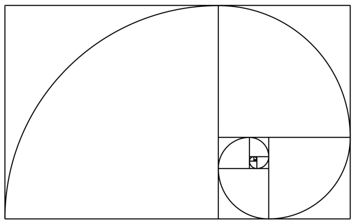
Sample Input 3

```
10
0 1
1 1
1 1
1 1
0 1
0 4
0 6
1 1
0 7
1 2
```

Sample Output 3

```
10 8 3 2 4
1 6 7 9 5
```

Problem F
Fibonacci Compression



Fibonacci compression is a new type of fault-tolerant compression based on Fibonacci numbers. Symbols are constructed according to the rule that no code word may have two consecutive "1" bits at any place other than the end, where they are mandatory. In practice this means that, for each compressed symbol bit-length i where $i \geq 2$, there are Fibonacci$(i-1)$ compressed symbols of that length.

For example, the shortest 14 Fibonacci code words are as follows:

```
11       011      0011    1011
00011    10011    01011   000011
100011   010011   001011  101011
0000011  1000011  ...
```

Compressing a string using Fibonacci compression works by replacing the most frequent characters with the shortest codes. Given one such string s, find the length of each of its prefixes when compressed as small as possible according to this system.

Input

- One line containing the length of the string to compress, n ($1 \leq n \leq 10^5$).
- One line containing the string s as a sequence of n integers s_i ($0 \leq s \leq 10^6$).

Output

Output $|s|$ lines, where the ith line is the compressed length of the first i characters of s, in bits.

Sample Input 1

```
4
97 97 98 98
```

Sample Output 1

```
2 4 7 10
```

Sample Input 2

```
24
1 75 2 1 1 75 75 75 75 75 75 2 2 3 4 5 6 7 8 9 10 11 12 10
```

Sample Output 2

```
2 5 9 11 13 16 19 21 23 25 27 31 35 39 44 49 54 60 66 72 78 84 91 95
```

Problem G
Garden Variety Vampire

Count Dracula is a voracious (as well as carnivorous) gardener. He keeps a small garden within easy walking distance of his eponymous mansion, and naturally also near to his eponymous potting shed.

Recently, the Count has had to cut back his gardening time sharply on account of adversely excellent weather conditions. His prize Deadly Nightshades are starting to feel neglected.

He would like to pave a shaded path between the garden, the mansion, and the shed using an ensemble of variously-sized trees he's obtained from Harker Nurseries at suspiciously low prices. Each tree casts a circle-shaped shadow that Dracula can walk across.

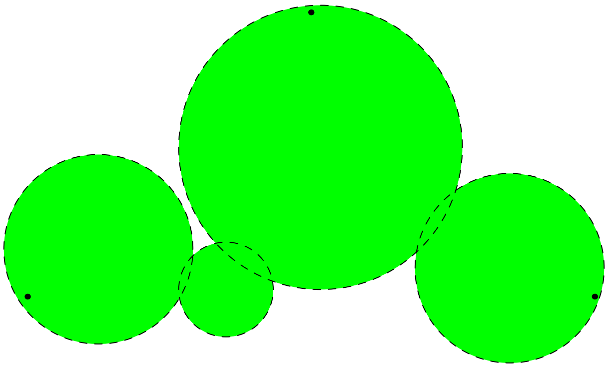


Figure G.1: Illustration of a solution to the second sample input. Dots mark the garden, mansion and shed.

Is it possible for him to re-plant his trees in such a way that they cast a continuous shadow on the ground connecting the three given locations? You may assume shadows are directly beneath the trees and do not move throughout the day.

Input

The input consists of:

- three lines each containing the integer co-ordinates of x_i and y_i, each the planar coordinates of one gardening location where ($-10^6 \leq x_i, y_i \leq 10^6$ for each i).
- one line with an integer n, ($1 \leq n \leq 12$): the number of trees the Count has acquired.
- one line containing n integers $s_1 \ldots s_n$ ($1 \leq s_i \leq 10^6$ for each i), the individual radii of the shadows cast by the Count's tree collection.

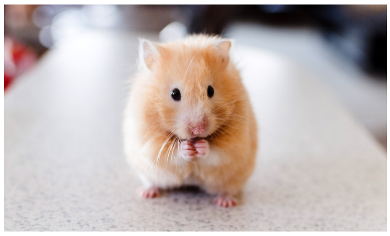
Output

If the Count can place the trees in such a way as to connect all of the sites, output "`possible`". Otherwise, output "`impossible`".

Sample Input 1	Sample Output 1
-6 0 6 0 0 6 4 1 3 1 1	impossible

Sample Input 2	Sample Output 2
-6 0 6 0 0 6 4 2 3 2 1	possible

Problem H
Hamster Ball



The years have been good to the Rock and Roll Hamster Ball company. Starting with a single hamster ball they now produce a popular range of different sizes for hamsters big and small.

Recently they moved manufacturing to a new plant but disaster has happened: The new balls, many of which have been shipped to customers, have a defect. The two halves of the ball do not lock together properly and hamsters are escaping.

The immediate solution is simple. We will send out special hamster ball tape to each customer, they put the hamster in the ball and tape it shut. Problem solved.

Unfortunately you only have a certain amount of tape and you're not sure it will seal all of the balls that have been shipped. Given the length of tape, the number and radius of hamster balls, what is the largest number of balls you can tape shut?

Input

- one line containing the integer t ($1 \le t \le 10000$), the length of the tape in centimetres.
- one line containing the integer b ($1 \le b \le 100$), the number of different sizes of balls.
- b lines each with integers d ($1 \le d \le 100$), the radius of a ball in centimeters and s ($1 \le s \le 10000$) the number of balls of that size sold

Output

A single line with the integer number of the largest number of balls you can tape shut. If you cannot tape shut any balls, output 0.

Sample Input 1

```
1000
2
2 30
3 20
```

Sample Output 1

```
5
```

Sample Input 2

```
2000
4
20 30
1 20
1 15
20 25
```

Sample Output 2

```
13
```

Problem I
Internet Upload

You made it! You tricked your way into the embassy and managed to copy the files your informant told you about. After a difficult escape, the authorities know what you've done and sealed off the city. You need to upload the data to the internet before they catch you.

Luckily the city is full of coffee shops with free wifi and you researched their details before you started your heist. The only downside is that they have strange opening hours and you may need to move between them. When travelling between coffee shops you can't upload any data at all.

Given the list of coffee shops, their opening hours and Wifi speed, what is the earliest time when all of your data can been uploaded?

Input

- The first line contains an integer d ($1 \leq d \leq 10^9$), the size of the data in megabytes.
- The second line contains an integer n ($1 \leq n \leq 100$), the number of cafes.
- The next n lines each contain:
  - The integer opening and closing times of the cafe in seconds, o and c ($0 \leq o \leq c \leq 24 * 60 * 60$).
  - The wifi speed of the cafe w ($1 \leq c \leq 1000$) in megabytes per second.
- n lines, each with n integers, the time to travel to each cafe from the n^{th} cafe, in seconds ($0 \leq d_i \leq 24 * 60 * 60$).

You may start in any single cafe when it opens and start using wifi the moment you arrive at a cafe.

Output

One line containing an integer number of seconds, the smallest integer time by which all of the data can be uploaded.

Sample Input 1	Sample Output 1
200 2 10 20 15 10 40 5 0 5 5 0	35

Problem J
Jackpot

The thunderingly fun new game show of ’Jackpot’ has bounded onto televisio screens up and down the land and is the next big thing. It runs along a simple premise: there are a number of doors, behind one of which is hidden a monetary prize and behind all of the others of which are hidden goats. Each contestant chooses how many doors will be opened before they pick a door to look at. The prize decreases as each door is opened. If they manage to open the door and find the money they get to keep it!

Not content with guessing, you’ve decided to try and work out the best number of doors to open to maximise your expected profit, where expected profit is the probability of winning multiplied by the remaining prize fund.

You’ve managed to find out that the remaining prize fund P_R is a function of the initial prize fund m, the number of doors opened d and a scaling factor f, ie.

$$P_R = m - (d \cdot f)^2$$

Given the number of doors, initial prize fund and the scaling factor can you work out how many doors should be opened before you try to pick a door to maximise your expected profit?

Input

One line of input containing:

- One integer n ($1 \leq n \leq 20,000$), the number of doors.
- One integer m ($1 \leq m \leq 10,000,000$) the initial prize fund.
- One floating point f ($0 < f \leq 1$), the scaling factor.

There will always only be one door with the best value.

Output

One line containing

- One floating point number, the value of the expected payout.

The output must be accurate to an absolute or relative error of at most 10^{-6}.

Sample Input 1

```
4 100 1
```

Sample Output 1

```
91.0
```

Sample Input 2

```
20000 100200300 0.8
```

Sample Output 2

```
5628.83998462
```

Problem K
Kings

A heated game of "chess-tidying" starts with a messy board containing several king pieces strewn across the cells. The player's job is to put all of the pieces into a line along the primary diagonal of the board, which runs along black squares.

In each move, unlike in normal chess, one king may be moved one place either horizontally or vertically (but not both) to another unoccupied cell.

Given an instance of the game, find how many moves you will need in order to finish it.

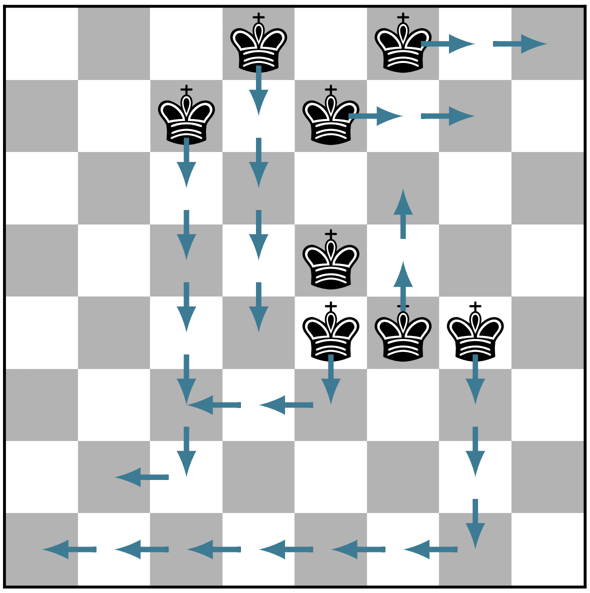


Figure K.1: Illustration of Sample Input 2. In this case, the minimum number of moves necessary to put all of the kings along the black diagonal is 28 as pictured.

Input

- One line with the number of rows and columns, n ($1 \leq n \leq 500$).
- Each of the following n lines contains the two-dimensional integer coordinates c and r ($1 \leq c, r \leq n$), the position of one of the kings.

Each of the kings starts at a unique position.

Output

Output the minimum number of moves necessary to cover the main diagonal ($r = c$) with kings.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

```
8
6 4
6 8
5 5
5 4
4 8
5 7
7 4
3 7
```

Sample Output 2

```
28
```

Problem L
Last Word

The `substring()` function is a commonly-used operation available in most programming languages that operates on strings. A start offset and a length are provided and used to construct a new string containing only the characters in a sequence of that length beginning from the offset.

One particular string has had this called a large number of times in sequence: we repeatedly used the standard library function `substring(s, start, length)` to chop it up until now a potentially much shorter string remains.

Find the value of the string produced by all of these operations.

Input

- The first line of input contains the string s ($1 \le |s| \le 10^6$).
- The second line of input contains the number of operations, n ($1 \le n \le 10^6$).
- Each of the following n lines contains the two integers $\texttt{start}_i$ and $\texttt{length}_i$ ($0 \le \texttt{start}_i < \texttt{length}_{i-1}; 1 \le \texttt{start}_i + \texttt{length}_i \le \texttt{length}_{i-1}$).

Output

Output the string after all of the successive substring() operations.

Sample Input 1

```
helloworld
2
1 9
0 5
```

Sample Output 1

```
ellow
```

Sample Input 2

```
abcdefghijklmnopqrstuvwxyz
8
1 24
1 22
1 20
1 18
1 16
1 14
1 12
1 10
```

Sample Output 2

```
ijklmnopqr
```