

CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)
Statement is not available
on English language

A. Sắc màu

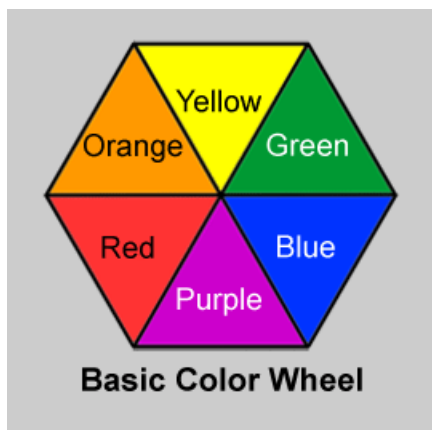
time limit per test: 1 second

memory limit per test: 256 megabytes

input: color.inp

output: color.out

Trong hội họa, *bánh xe sắc màu* là một khái niệm nói về việc chia các màu sắc thành những màu cơ bản, và tạo những màu khác bằng cách pha trộn các màu cơ bản với nhau. Một trong những bánh xe sắc màu được thể hiện như hình dưới đây:



Có ba màu cơ bản là đỏ, vàng và xanh lam. Trộn màu đỏ và màu vàng sẽ cho ra màu da cam. Trộn màu vàng và xanh lam cho ra xanh lục. Trộn xanh lam và đỏ cho ra màu tím. Có ba cặp màu tương phản là đỏ và xanh lục, da cam và xanh lam, vàng và tím.

Trong bài này, bạn được cho hai màu, hãy xác định xem chúng kề nhau trên bánh xe màu, là cặp màu tương phản hay sẽ trộn ra màu gì khác.

Input

Gồm hai dòng, mỗi dòng chứa một trong các từ Red, Orange, Yellow, Green, Blue và Violet mô tả một màu.

Output

- Nếu hai màu đã cho giống nhau, in ra từ *Same*.
- Nếu hai màu đã cho là hai màu kề nhau trên bánh xe màu sắc, in ra từ *Adjacent*.
- Nếu hai màu đã cho nằm hai bên của một màu khác trên bánh xe màu sắc, in ra tên của màu ở giữa.
- Nếu hai màu đã cho là hai màu tương phản, in ra từ *Complementary*.
- In ra *None* trong các trường hợp còn lại.

Examples

input

Copy

Red
Red**Cute bubbletea lovers 2022**

Participant

[→ About Group](#)

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:31:27

Contestant

[→ Languages](#)

The following languages are only available languages for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

[→ Submit?](#)

Language: GNU G++14 6.4.0

 Choose file: No file chosen

Be careful: the problem requires input/output via file(s).

[→ Last submissions](#)

Submission	Time	Verdict

↑	output	Copy
	Same	
	input	Copy
	Red Orange	
	output	Copy
	Adjacent	
	input	Copy
	Red Yellow	
	output	Copy
	Orange	
	input	Copy
	Red Green	
	output	Copy
	Complementary	

161620251	Jun/24/2022 11:45	Perfect result: 100 points
---------------------------	----------------------	---

→ Your points

	Points
A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
 The only programming contests Web 2.0 platform
 Server time: Jun/25/2022 19:03:26^{UTC+7} (i2).
 Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by



CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)

 Statement is not available
on English language

B. Số nguyên tố đối xứng

time limit per test: 0.5 seconds

memory limit per test: 32 megabytes

input: palinprime.inp

output: palinprime.out

Một số nguyên dương được gọi là số nguyên tố khi và chỉ khi số này có đúng hai ước số nguyên dương là 1 và chính nó.

Một số nguyên dương được gọi là số đối xứng khi và chỉ khi nếu như coi dãy chữ số biểu diễn số này trong hệ thập phân là một xâu kí tự, thì xâu kí tự này là một xâu đối xứng (tức đọc từ trái sang phải và đọc từ phải qua trái cho ra cùng một xâu).

Cho hai số nguyên dương l và r , hãy tính tổng các số nguyên tố đối xứng trong đoạn từ l đến r .

Input

Gồm một dòng duy nhất chứa hai số nguyên l và r ($1 \leq l \leq r \leq 10^7$).

Output

In ra một số nguyên duy nhất là tổng các số nguyên tố đối xứng trong đoạn từ l đến r .

Scoring

- Subtask 1 (30 điểm): $r \leq 1000$
- Subtask 2 (70 điểm): $r \leq 10000000$

Examples

input	Copy
100 150	
output	Copy
232	

input	Copy
1 9	
output	Copy
17	

Note

Trong ví dụ thứ nhất, các số nguyên tố đối xứng có giá trị từ 2 đến 9 là 2, 3, 5 và 7.

Trong ví dụ thứ hai, các số nguyên tố đối xứng có giá trị từ 100 đến 150 là 101 và 131.

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:31:00

Contestant

→ Languages

The following languages are only available languages for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language: GNU G++14 6.4.0 [v](#)
 Choose file: [Choose File](#) No file chosen

Be careful: the problem requires input/output via file(s).

[Submit](#)

→ Last submissions

Submission	Time	Verdict



161713294	Jun/25/2022 12:51	Partial result: 34 points
161712426	Jun/25/2022 12:40	Partial result: 44 points
161712175	Jun/25/2022 12:37	Partial result: 70 points
161711979	Jun/25/2022 12:35	Partial result: 66 points
161711801	Jun/25/2022 12:32	Partial result: 58 points
161711711	Jun/25/2022 12:31	Partial result: 16 points
161627066	Jun/24/2022 13:11	Partial result: 42 points
161626961	Jun/24/2022 13:09	Partial result: 0 points
161624381	Jun/24/2022 12:36	Partial result: 54 points
161624176	Jun/24/2022 12:34	Partial result: 2 points

→ **Your points**

	Points
A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
 The only programming contests Web 2.0 platform
 Server time: Jun/25/2022 19:03:57^{UTC+7} (i2).
 Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by





CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)
Statement is not available
on English language

C. Ước số thực sự

time limit per test: 0.25 seconds
 memory limit per test: 64 megabytes
 input: `divexp.inp`
 output: `divexp.out`

Cho số nguyên dương k . Ta gọi số nguyên dương d là *ước số thực sự* của số nguyên dương n khi và chỉ khi d chia hết n nhưng d^k không chia hết n . Ví dụ, nếu $k = 2$; 4, 20, 25, 50 và 100 là các *ước số thực sự* của 100; nhưng 1, 2, 5 hay 10 thì không.

Cho hai số nguyên dương l và r . Gọi $\alpha(x)$ là số *ước số thực sự* của số nguyên dương x , hãy tính $\alpha(l) + \alpha(l+1) + \dots + \alpha(r-1) + \alpha(r)$.

Input

Gồm một dòng với ba số nguyên l , r và k ($1 \leq l \leq r \leq 10^7$, $2 \leq k \leq 10$).

Output

In ra một số nguyên duy nhất là giá trị cần tìm.

Scoring

- Subtask 1 (37 điểm): $r \leq 1000$
- Subtask 2 (63 điểm): $r \leq 10000000$

Example

input	Copy
100 102 2	
output	Copy
13	

Note

- Số 100 có 5 *ước số thực sự* là 4, 20, 25, 50 và 100.
- Số 101 có 1 *ước số thực sự* là 101.
- Số 102 có 7 *ước số thực sự* là 2, 3, 6, 17, 34, 51 và 102.

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:30:26

Contestant

→ Languages

The following languages are only available for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language: GNU G++14 6.4.0 ▼

Choose file: Choose File No file chosen

Be careful: the problem requires input/output via file(s).

Submit

→ Last submissions

Submission	Time	Verdict



161685281	Jun/25/2022 04:45	Perfect result: 100 points
161685133	Jun/25/2022 04:40	Partial result: 99 points
161684487	Jun/25/2022 04:19	Partial result: 28 points
161684160	Jun/25/2022 04:08	Partial result: 28 points

→ Your points

	Points
A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jun/25/2022 19:04:29^{UTC+7} (i2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by



CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)
Statement is not available
on English language

D. Ước số lẻ

time limit per test: 0.25 seconds
 memory limit per test: 32 megabytes
 input: odddiv.inp
 output: odddiv.out

Với một số nguyên dương n bất kì, ta gọi $\gamma(n)$ là ước nguyên dương lẻ lớn nhất của số n . Ví dụ $\gamma(2) = 1$, $\gamma(3) = 3$, $\gamma(4) = 1, \dots$

Cho hai số nguyên dương l và r , hãy tính $\gamma(l) + \gamma(l+1) + \dots + \gamma(r)$.

Input

Gồm một dòng với hai số nguyên l và r ($1 \leq l \leq r \leq 10^9$).

Output

In ra một số nguyên duy nhất là giá trị cần tìm.

Scoring

- Subtask 1 (26 điểm): $r \leq 10^3$
- Subtask 2 (36 điểm): $r \leq 10^6$
- Subtask 3 (38 điểm): $r \leq 10^9$

Examples

input	Copy
2 5	
output	Copy
10	

input	Copy
1 1	
output	Copy
1	

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:30:00

Contestant

→ Languages

The following languages are only available languages for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language: GNU G++14 6.4.0 ▼

 Choose file: [Choose File](#) No file chosen

Be careful: the problem requires input/output via file(s).

[Submit](#)

→ Last submissions

Submission	Time	Verdict



161711016	Jun/25/2022 12:22	Perfect result: 100 points
161710930	Jun/25/2022 12:22	Partial result: 98 points
161705068	Jun/25/2022 11:07	Partial result: 32 points
161686414	Jun/25/2022 05:15	Partial result: 15 points
161686350	Jun/25/2022 05:13	Partial result: 8 points
161686284	Jun/25/2022 05:12	Partial result: 0 points

→ Your points

	Points
A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jun/25/2022 19:04:59^{UTC+7} (i2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by





CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)

 Statement is not available
 on English language

E. Ước lũy thừa

 time limit per test: 0.25 seconds
 memory limit per test: 32 megabytes
 input: divpow.inp
 output: divpow.out

Với một số nguyên dương η bất kì, ta gọi $\delta(\eta)$ là số ước số nguyên dương của số η . Tiếp theo, ta kí hiệu $\mu(\eta) = \eta^{\delta(\eta)}$.

Ví dụ, số 6 có các ước nguyên dương là 1, 2, 3 và 6. Do đó $\mu(6) = 6^4 = 1296$.

Cho số nguyên dương ξ , tìm số nguyên dương η nhỏ nhất để $\mu(\eta) = \xi$.

Input

Gồm một số nguyên ξ duy nhất ($1 \leq \xi \leq 10^{18}$).

Output

In ra số nguyên dương η nhỏ nhất thỏa mãn $\mu(\eta) = \xi$. Nếu không có số nào như vậy, in ra -1.

Scoring

- Subtask 1 (24 điểm): $\xi \leq 10^6$
- Subtask 2 (76 điểm): $\xi \leq 10^{18}$

Example

input	Copy
1296	
output	Copy
6	

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

**2022 summer contest #20, DN
 Camp round 2 (for group A2).**

Contest is running

01:29:29

Contestant

→ Languages

The following languages are only available
 languages for the problems from the
 contest

**2022 summer contest #20, DN Camp
 round 2 (for group A2):**

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language: GNU G++14 6.4.0 ▼

Choose file: Choose File No file chosen

**Be careful: the problem requires
input/output via file(s).**

Submit

→ Your points

	Points



A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jun/25/2022 19:05:30^{UTC+7} (i2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by





CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)
Statement is not available
on English language

F. Ước số chung lớn nhất - bội số chung nhỏ nhất

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

input: gcdlcm.inp

output: gcdlcm.out

Trong một bài giảng môn Số học của mình, GSPVH giới thiệu cho các học sinh về ước số chung lớn nhất và bội số chung nhỏ nhất. Cụ thể, GS đã nói như sau: *Ước số và bội số là hai khái niệm vô cùng quen thuộc trong số học. Với hai số nguyên a và b bất kỳ, nếu a chia hết cho b , ta nói a là bội số của b và b là ước số của a . Với một bộ k số nguyên dương x_1, x_2, \dots, x_k bất kỳ, ước số chung lớn nhất của chúng là số nguyên dương y lớn nhất thỏa mãn x_i là bội số của y với mọi $1 \leq i \leq k$. Tương tự, bội số chung nhỏ nhất của chúng là số nguyên dương z nhỏ nhất thỏa mãn x_j là ước số của z với mọi $1 \leq j \leq k$.*

Tuy nhiên, khi nói đến đây, GSPVH nhận ra có một số học sinh trong lớp không thực sự chú ý nghe giảng. Thay vào đó, nhiều bạn mãi nói chuyện riêng, nhắn tin cho gấu, lướt Facebook, ăn cơm trễ,... Vì vậy, để kiểm tra mức độ hiểu bài của các bạn học sinh, GSPVH đã giao một bài toán như sau:

Cho hai dãy số nguyên dương p_1, p_2, \dots, p_m và q_1, q_2, \dots, q_n và một số nguyên dương k . Đặt $P = p_1 \cdot p_2 \cdot \dots \cdot p_m$ và $Q = q_1 \cdot q_2 \cdot \dots \cdot q_n$. Hãy đếm số bộ k số nguyên dương (r_1, r_2, \dots, r_k) sao cho chúng có ước số chung lớn nhất là P và bội số chung nhỏ nhất là Q .

Do kết quả có thể rất bé, bạn chỉ cần in ra số bộ số thỏa mãn modulo 998244353.

Input

Dòng đầu tiên chứa ba số nguyên m, n và k ($1 \leq m, n, \sqrt[3]{k^2} \leq 10^6$).

Dòng thứ hai chứa m số nguyên p_1, p_2, \dots, p_m ($1 \leq p_i \leq 15 \cdot 10^5$).

Dòng thứ ba chứa n số nguyên q_1, q_2, \dots, q_n ($1 \leq q_j \leq 15 \cdot 10^5$).

Output

In ra một số nguyên duy nhất là số bộ số thỏa mãn modulo 998244353.

Scoring

- Subtask 1 (10 điểm): $m = n = k = 1$
- Subtask 2 (18 điểm): $1 \leq P, Q \leq 15 \cdot 10^5$ và $k = 2$
- Subtask 3 (22 điểm): $m, n \leq 5000$ và $k = 2$
- Subtask 4 (22 điểm): $k = 2$
- Subtask 5 (28 điểm): Không có ràng buộc gì thêm.

Examples

input

Copy

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

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:28:57

Contestant

→ Languages

The following languages are only available for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language: GNU G++14 6.4.0

Choose file: No file chosen

Be careful: the problem requires input/output via file(s).

→ Your points

Points

	output	Copy
	4	
	input	Copy
	2 1 10 1 3 9	
	output	Copy
	1022	
	input	Copy
	7 4 1962 2 2 7 1 9 9 7 22 7 19 97	
	output	Copy
	0	

A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Note

Trong ví dụ đầu tiên, ta có $P = 1$ và $Q = 6$. Các bộ số thỏa mãn là $(1, 6)$, $(2, 3)$, $(3, 2)$ và $(6, 1)$.

Trong ví dụ thứ hai, ta có $P = 3$ và $Q = 9$. Bộ 10 số thỏa mãn là bộ chỉ gồm các số 3 và 9, và phải có ít nhất một số 3 và ít nhất một số 9. Do đó số bộ số thỏa mãn là $2^{10} - 2 = 1022$.

Trong ví dụ thứ ba, ta có $P = 15876$ và $Q = 283822$. Do Q không chia hết cho P , không có bộ số nào thỏa mãn.

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
 The only programming contests Web 2.0 platform
 Server time: Jun/25/2022 19:06:01^{UTC+7} (i2).
 Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by



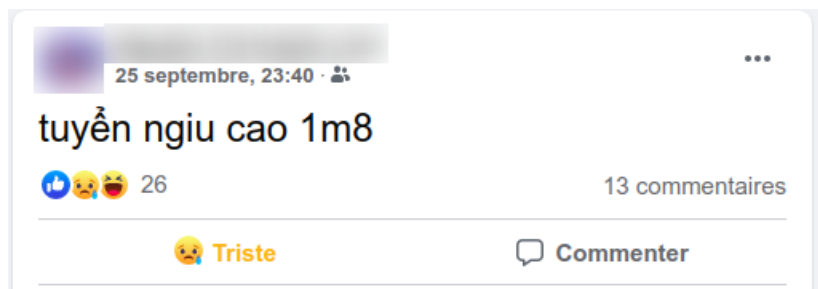
CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)
Statement is not available
on English language

G. Tuyển người yêu

 time limit per test: 0.75 seconds
 memory limit per test: 256 megabytes
 input: couples.inp
 output: couples.out

Nếu dành vài giờ mỗi ngày để lướt newsfeed trên Facebook, các bạn không khó để tìm ra những status thả thính tuyển người yêu công khai như thế này. Các bạn trẻ khi tuyển người yêu thường đưa ra hàng loạt các tiêu chuẩn đa dạng phong phú, như chỉ cần nặng trên 90kg, chấp nhận đeo bạn gái đi uống trà sữa ít nhất 3 lần mỗi ngày. Một số bạn nữ yêu thơ thì còn viết tiêu chuẩn bạn trai theo cách đầy vần điệu như sau: "Không giàu thì phải đẹp trai. Không thông kinh sử phải dài... cái chân". Nhưng có một tiêu chí chọn bạn trai, bạn gái xuất hiện nhiều hơn cả, đó chính là chiều cao.



Nắm được nhu cầu thoát nạn FA rất lớn của các trẻ trâu viên, Hoàng Thế Anh — chủ tịch CEC (viết tắt của chữ *Club of Excited Couples*, tạm dịch là Câu lạc bộ các cặp đôi tăng động), tổ chức sự kiện giúp các trẻ trâu trai gái có cơ hội gặp mặt và tìm lấy nửa kia cho mình.

Tham gia sự kiện lần này, có m bạn nam và f bạn nữ tham gia. Mỗi bạn khi đến sự kiện đều cho biết chiều cao và gu người yêu của mình. Gu của mỗi bạn luôn thuộc một trong hai dạng, người yêu của mình phải thấp hơn hoặc cao hơn mình. Kỳ lạ thay, không ai muốn yêu một người có chiều cao chính xác bằng mình.

Sau khi nhận hồ sơ và duyệt đơn, Thế Anh ghép cặp các bạn trai và các bạn gái với nhau, để rồi hai người bước vào những ngày hẹn hò thâm kín. Để tránh sự giao động, hụt hẫng hoặc các phi vụ đánh ghen có thể xảy ra, không có bạn trai hay bạn gái nào được hẹn hò với nhiều hơn một người khác giới, và các bạn được ghép cặp phải thỏa mãn gu chiều cao của cả hai người. Cho biết chiều cao và khẩu vị của các bạn tham gia, hãy giúp Thế Anh ghép cặp sao cho có nhiều đôi hẹn hò nhất.

Input

- Dòng đầu tiên chứa số nguyên θ ($1 \leq \theta \leq 4$) — số thứ tự của subtask chứa test này.
- Dòng thứ hai chứa số nguyên m ($1 \leq m \leq 3 \cdot 10^5$) — số bạn nam tham gia tìm bạn gái.
- Dòng thứ ba chứa m số nguyên b_1, b_2, \dots, b_m ($1 \leq |b_i| \leq 10^9$), trong đó $|b_i|$ là chiều cao của bạn nam thứ i . Nếu $b_i < 0$, bạn này muốn có bạn gái thấp hơn mình. Nếu $b_i > 0$, bạn này muốn có bạn gái cao hơn mình.
- Dòng thứ tư chứa số nguyên f ($1 \leq f \leq 3 \cdot 10^5$) — số bạn nữ tham gia tìm bạn trai.

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:28:27

Contestant

→ Languages

The following languages are only available for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language: GNU G++14 6.4.0

 Choose file: No file chosen

Be careful: the problem requires input/output via file(s).

→ Your points

Points



- Dòng thứ năm chứa f số nguyên g_1, g_2, \dots, g_f ($1 \leq |g_i| \leq 10^9$), thể hiện thông tin về chiều cao và gu bạn trai của các bạn nữ này, theo cách tương tự như dãy b_1, b_2, \dots, b_m .

Output

Dòng đầu tiên chứa số nguyên k là số cặp đôi tối đa mà Thế Anh có thể ghép được

k dòng tiếp theo, mỗi dòng chứa hai số nguyên x và y ($1 \leq x \leq m, 1 \leq y \leq f$) cho biết bạn nam x được ghép đôi với bạn nữ y .

Nếu có nhiều đáp án tối ưu, bạn được phép in ra một phương án bất kì.

Scoring

- Subtask 1 (25 điểm): $m, f \leq 10$
- Subtask 2 (25 điểm): $1 \leq |b_i|, |g_j| \leq 2$ với mọi i và j .
- Subtask 3 (25 điểm): $1 \leq |b_i|, |g_j| \leq 200$ với mọi i và j .
- Subtask 4 (25 điểm): Không có ràng buộc gì thêm.

Examples

input	Copy
1 1 227 1 -1997	
output	Copy
1 1 1	

input	Copy
1 2 -100 200 3 60 100 -300	
output	Copy
2 2 3 1 1	

A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jun/25/2022 19:06:32^{UTC+7} (i2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Powered by



CONTESTS

[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [STANDINGS](#) [CUSTOM INVOCATION](#)
Statement is not available
on English language

H. Bi đổi màu

time limit per test: 1.25 seconds
memory limit per test: 256 megabytes
input: bidoimau.inp
output: bidoimau.out

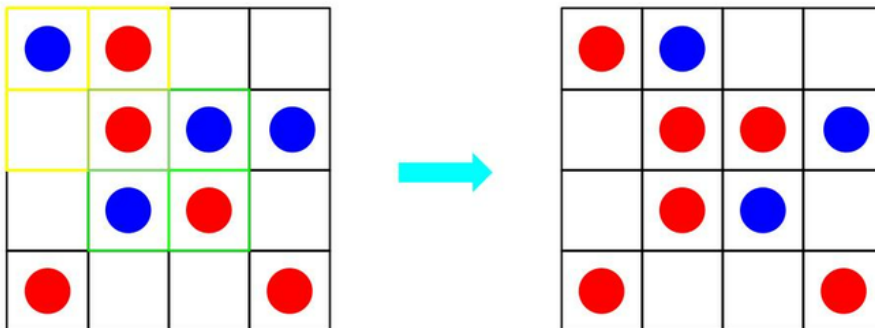
Trò chơi "bi đổi màu" là trò chơi đối kháng hai người trên lưới hình chữ nhật. Cụ thể, hai người sẽ chơi trên một bảng gồm m hàng và n cột. Các hàng được đánh số từ 1 đến m theo thứ tự từ trên xuống dưới, các cột được đánh số từ 1 đến n theo thứ tự từ trái qua phải. Ô nằm ở giao của hàng i và cột j được kí hiệu là (i, j) . Ban đầu, trên bảng có một số viên bi đỏ và một số viên bi xanh. Một ô có thể không chứa viên bi nào nhưng không thể chứa nhiều hơn một viên bi.

Trò chơi diễn ra trong một số lượt. Tại mỗi lượt chơi, mỗi người chơi sẽ chọn ra một hình vuông con có kích thước $p \times p$ của bảng và bí mật viết ra giấy. Sau đó, trọng tài sẽ thu lại các mẫu giấy và công bố sự lựa chọn của hai người. Dựa trên các sự lựa chọn này, màu của các viên bi trên bảng sẽ được cập nhật lại như sau: Một viên bi bị đổi màu (từ xanh sang đỏ hoặc ngược lại) khi và chỉ khi viên bi này thuộc **chính xác** một hình vuông được lựa chọn bởi một trong hai người chơi. Các viên bi không thuộc hình vuông con nào hoặc thuộc cả hai hình vuông con đều không bị đổi màu.

Mục tiêu của người chơi thứ nhất là cực đại hóa số bi xanh trên bảng khi trò chơi kết thúc, mục tiêu của người còn lại là cực đại hóa số bi đỏ trên bảng khi trò chơi kết thúc.

(Phỏng theo đề thi tin học trẻ toàn quốc bằng C, năm 2012)

Ví dụ, giả sử trạng thái của trò chơi trước một lượt chơi giống như ở hình bên trái. Khi đó, nếu người chơi thứ nhất chọn hình vuông 2×2 có góc trái trên là $(1, 1)$ (chứa các ô màu vàng) và người chơi thứ hai chọn hình vuông có góc trái trên là $(2, 2)$ (chứa các ô màu xanh lục), các viên bi sẽ thay đổi màu như hình bên phải.



Do không được biết sự lựa chọn của đối phương, việc tìm ra một nước đi tốt là không hề đơn giản. Chẳng hạn, với người chơi thứ nhất, việc chọn một hình vuông con chứa nhiều viên bi màu đỏ (với hi vọng đổi màu chúng sang xanh) chưa chắc đã tốt vì có thể bị đối phương khắc chế bằng cách chọn một vùng khác gồm toàn bi xanh. Do đó, nếu giả sử hai người đều chơi

Cute bubbletea lovers 2022

Participant

→ About Group

Nơi hội tụ của những người cute và đam mê trà sữa. Nơi trao đổi, giao lưu, hẹn hò của những fan cuồng trà sữa trung thành nhất.

2022 summer contest #20, DN Camp round 2 (for group A2).

Contest is running

01:27:55

Contestant

→ Languages

The following languages are only available for the problems from the contest

2022 summer contest #20, DN Camp round 2 (for group A2):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)
- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.4.2

→ Submit?

Language:

Choose file: No file chosen

Be careful: the problem requires input/output via file(s).

→ Your points

	Points

↑ tối ưu, độ tốt của một nước đi với người thứ nhất được tính bằng số viên bi xanh có trên bảng trong trường hợp xấu nhất, tức số viên bi xanh nhỏ nhất có thể có khi xét mọi phương án lựa chọn của đối phương.

Nói cách khác, gọi $\sigma(x, y, i, j)$ là số viên bi xanh trên bảng sau lượt chơi nếu người chơi thứ nhất chọn hình vuông với góc trái trên ở ô (x, y) và người chơi thứ hai chọn hình vuông với góc trái trên ở ô (i, j) ($1 \leq x, i \leq m - p + 1, 1 \leq y, j \leq n - p + 1$), độ tốt của nước đi "chọn hình vuông với góc trái trên ở ô (x, y) " đối với người chơi thứ nhất được tính như sau:
 $\mu(x, y) = \min \sigma(x, y, i, j)$ với mọi $1 \leq i \leq m - p + 1$ và $1 \leq j \leq n - p + 1$.

Cho biết trạng thái của bảng trước một lượt chơi, hãy tính độ tốt của tất cả các nước đi có thể của người chơi thứ nhất.

Input

Dòng đầu tiên chứa ba số nguyên m, n và p ($1 \leq m, n \leq 500, 1 \leq p \leq \min(m, n, 20)$), lần lượt là số hàng, số cột của bảng và kích thước của vùng hình vuông được chọn để đổi màu.

Trong m dòng còn lại, mỗi dòng chứa một xâu độ dài n chỉ gồm các kí tự `.`, `r` và `b` thể hiện trạng thái của bảng trước lượt chơi. Ô (i, j) chứa một viên bi màu đỏ khi và chỉ khi kí tự thứ j của xâu thứ i là `r`. Ô này chứa một viên bi màu xanh khi và chỉ khi kí tự tương ứng là `b`.

Output

In ra $m - p + 1$ dòng, mỗi dòng chứa $n - p + 1$ số nguyên. Số thứ y trên dòng thứ x là giá trị $\mu(x, y)$.

Scoring

- Subtask 1 (22 điểm): $m, n \leq 30$
- Subtask 2 (36 điểm): $m, n \leq 100$
- Subtask 3 (42 điểm): $m, n \leq 500$

Example

input	Copy
<pre>4 4 2 br.. .rbb .br. r..r</pre>	
output	Copy
<pre>2 2 2 2 3 1 2 1 3</pre>	

A	100
B	70
C	100
D	100
E	
F	
G	
H	
Total	370

B10-P

Tóm tắt đề bài:

Có một bảng gồm m hàng và n cột. Trên mỗi ô có tối đa một viên bi. Mỗi viên bi có thể là màu xanh hoặc màu đỏ. Hai người chơi, mỗi người chọn ra một hình vuông con của bảng có kích thước $p \times p$. Sau đó, các viên bi trên bảng sẽ đổi màu dựa theo sự lựa chọn của hai người:

- + Viên bi nằm trên chính xác một trong hai hình vuông $p \times p$ sẽ đổi màu (đỏ \rightarrow xanh hoặc xanh \rightarrow đỏ)

- + Viên bi nằm trên cả hai hình vuông hoặc không nằm trên bất kì hình vuông nào sẽ giữ nguyên màu.

Người A muốn số bi xanh là nhiều nhất có thể. Người B muốn số bi đỏ là nhiều nhất có thể (tương đương với số bi xanh ít nhất).

Người A muốn biết, giả sử mình chọn hình vuông $p \times p$ có góc trái trên là (x, y) thì số viên bi xanh trên bàn nhỏ nhất là bao nhiêu.

Chú ý: một hình vuông kích thước $p \times p$ trong bảng m hàng n cột được xác định duy nhất bởi tọa độ góc trái trên (x, y) thỏa mãn $1 \leq x \leq m - p + 1$ và $1 \leq y \leq n - p + 1$. Khi đó, hình vuông này chứa tất cả các ô (i, j) thỏa mãn $x \leq i \leq x + p - 1$ và $y \leq j \leq y + p - 1$

Subtask 1

Mô phỏng: với mỗi nước đi (x, y) của người thứ nhất, ta xét tất cả các cách đi (i, j) của người thứ hai. Với mỗi cặp sự lựa chọn của hai người, ta đổi màu các viên bi trên bảng và đếm số bi xanh sau biến đổi.

Chú ý: trong phần mô phỏng, thay vì làm theo đề bài “những viên bi nằm trên cả hai vùng chọn sẽ giữ nguyên màu”, ta hiểu rằng những viên bi này bị đổi màu hai lần, và do đó quay trở về màu ban đầu. Do đó, tất cả các viên bi nằm trong vùng chọn của người thứ nhất sẽ đổi màu một lần, và các viên bi nằm trong vùng chọn của người thứ hai sẽ đổi màu thêm lần nữa.

```
#define MAX    505
const int INF = (int)1e9 + 7;

char board[MAX][MAX]; // trạng thái bảng
int numRow, numCol, sz; // số hàng, số cột của bảng và kích thước vùng chọn

void input(void) {
    scanf("%d%d%d", &numRow, &numCol, &sz);
    for (int i = 1; i <= numRow; i++) scanf("%s", board[i] + 1);
}

// trả về số lượng bóng màu xanh
int countBlue(void) {
    int res = 0;
    for (int i = 1; i <= numRow; i++) for (int j = 1; j <= numCol; j++)
        if (board[i][j] == 'b') res++;
    return res;
}

// đổi màu tất cả các bi trong hình vuông p*p có góc trái trên là (x, y)
```

```

void update(int x, int y) {
    for (int i = 0; i < sz; i++) for (int j = 0; j < sz; j++)
        if (board[x + i][y + j] != '.') board[x + i][y + j] ^= 'r' ^ 'b';
}

// trường hợp tệ nhất nếu người thứ nhất chọn hình vuông có góc trái trên là (x, y).
int calcResult(int x, int y) {
    int res = INF;
    // xét tất cả các cách đi của người thứ hai:
    for (int i = 1; i <= numRow - sz + 1; i++) for (int j = 1; j <= numCol - sz + 1; j++)
        // cập nhật bảng theo hai sự lựa chọn là (x, y) và (i, j)
        update(x, y); update(i, j);
    minimize(res, countBlue());
    // sau khi ghi nhận kết quả, ta cần cập nhật lại, bản chất là
    // trả về bảng của trạng thái ban đầu (đọc vào từ input)
    update(x, y); update(i, j);

    return res;
}

// xét các nước đi của người chơi thứ nhất và tính kết quả
for (int x = 1; x <= numRow - sz + 1; x++) {
    for (int y = 1; y <= numCol - sz + 1; y++) printf("%d ", calcResult(x, y));
    printf("\n");
}

```

Giả sử có một biến x hiện đang chắc chắn nhận một trong hai giá trị a và b , khi đó lệnh gán $x \text{ ^= } a \text{ ^ } b$ sẽ đưa x về giá trị ngược lại.

$x \text{ ^= } a \text{ ^ } b$ tương đương với $\text{if } (x == a) \ x = b; \text{ else } x = a;$

Ví dụ:

+ Nếu hiện tại $x = 5$, $x \text{ ^= } 10 \text{ ^ } 5 \rightarrow x = 10$

+ Nếu hiện tại $x = 10$, $x \text{ ^= } 10 \text{ ^ } 5 \rightarrow x = 5$

Giải thích: chú ý rằng phép xor có các tính chất giao hoán và kết hợp tương tự như phép cộng và phép nhân.

Đồng thời, một số xor với chính nó $= 0$, một số xor với $0 =$ chính nó.

Do đó, lệnh $x \text{ ^= } a \text{ ^ } b$ nghĩa là gán $x = x \text{ ^ } a \text{ ^ } b$.

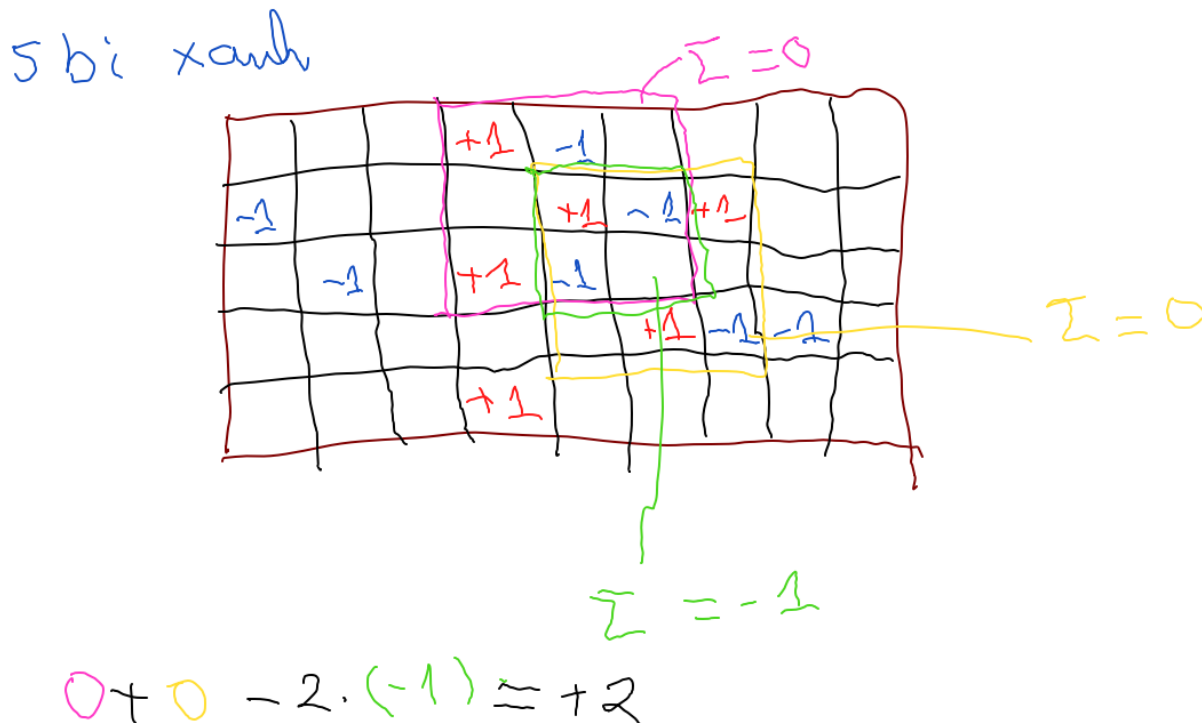
Nếu $x = a$, $x \text{ ^ } a \text{ ^ } b = a \text{ ^ } a \text{ ^ } b = 0 \text{ ^ } b = b$

Nếu $x = b$, $x \text{ ^ } a \text{ ^ } b = b \text{ ^ } a \text{ ^ } b = b \text{ ^ } b \text{ ^ } a = 0 \text{ ^ } a = a$

Subtask 2

Khi $m, n \leq 100$; ta vẫn duyệt tất cả các cặp nước đi của hai người chơi. Nói cách khác, ta duyệt tất cả các bộ 4 (x, y, i, j) với ý nghĩa: người thứ nhất chọn vùng có góc trái trên là (x, y) và người thứ hai chọn vùng có góc trái trên là (i, j) . Chi phí duyệt toàn bộ các bộ 4 này là $O(m^2 * n^2)$. Do đó, ta không thể mô phỏng việc đổi màu của các viên bi và tính số bi xanh bằng tay. Thay vào đó, ta sử dụng mảng tổng dồn để tính nhanh số bi xanh sau biến đổi trong $O(1)$.

Để tính nhanh số bi xanh sau khi biến đổi, ta tưởng tượng như sau. Ta gọi số bi xanh trên bảng là “số điểm” mà người chơi nhận được. Khi đó, việc đổi màu một viên bi từ xanh sang đỏ làm người chơi mất đi một điểm và việc đổi màu một viên bi từ đỏ sang xanh làm người chơi tăng thêm một điểm. Như vậy, ta có thể hiểu rằng một viên bi màu đỏ có giá trị là +1 và viên bi màu xanh có giá trị là -1. Khi đó, để tính lại số bi xanh trong bảng sau khi có vùng bị đổi màu, ta lấy số bi xanh ban đầu + tổng giá trị của các ô trong vùng bị đổi màu. Việc tính tổng giá trị của các ô trong một hình chữ nhật con bất kì có thể thực hiện trong $O(1)$ bằng mảng tổng dồn 2D.



Nếu hai vùng chọn (hình vuông $p \times p$) của hai người chơi không giao nhau, ta chỉ cần lấy tổng số điểm của các ô trong cả hai vùng thì sẽ tính được số bi xanh sau khi đổi màu. Tuy nhiên, khi hai vùng này có phần chung, cách tính trên không còn chính xác. Thực tế, việc đổi màu bi chỉ diễn ra ở những ô thuộc chính xác một trong hai vùng chọn, còn hoàn toàn không có ở phần giao. Như vậy, ta chỉ được điểm ở những ô thuộc chính xác một trong hai vùng còn không được điểm ở phần giao. Nếu tính tổng điểm trong vùng chọn của người 1 rồi cộng với tổng điểm trong vùng chọn của người 2, thì phần giao này đã được cộng vào 2 lần. Để loại trừ điểm ở phần giao, ta cần tính điểm của các ô trong phần giao và trừ đi 2 lần tổng các ô trong này.

```
char board[MAX][MAX]; // trạng thái bảng
int numRows, numCol, sz, cntBlue; // cntBlue = số bóng xanh ban đầu
int sum[MAX][MAX]; // mảng tổng dồn cho việc tính điểm

void input(void) {
    scanf("%d%d%d", &numRow, &numCol, &sz);
    for (int i = 1; i <= numRows; i++) scanf("%s", board[i] + 1);
    for (int i = 1; i <= numRows; i++) for (int j = 1; j <= numCol; j++) {
        if (board[i][j] == 'b') cntBlue++;
        sum[i][j] = sum[i][j - 1] + sum[i - 1][j] - sum[i - 1][j - 1]
            + (board[i][j] == 'r') - (board[i][j] == 'b');
    }
}
```

```

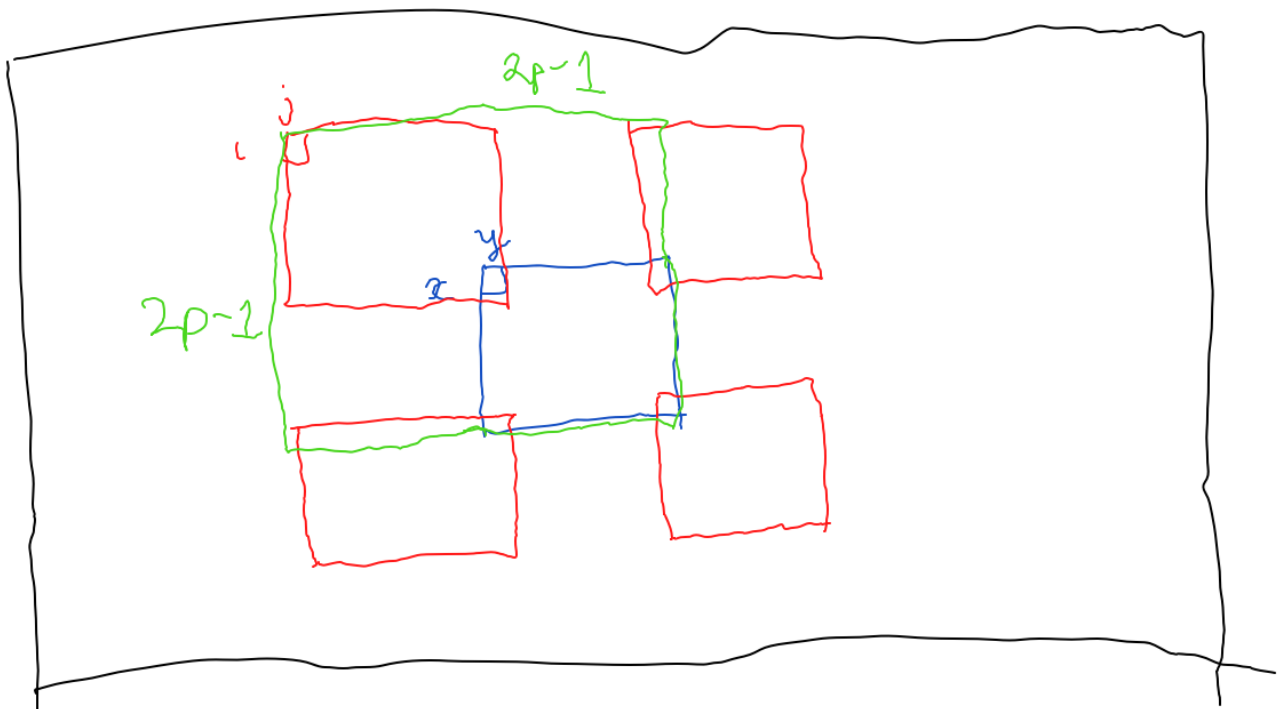
// trả về tổng số điểm các ô trong hình chữ nhật con từ hàng top đến hàng bot,
// từ cột left đến cột right
int getSum(int top, int bot, int left, int right) {
    // vùng không hợp lệ
    if (top > bot) return 0;
    if (left > right) return 0;
    return sum[bot][right] - sum[top - 1][right] - sum[bot][left - 1]
        + sum[top - 1][left - 1];
}

// trả về số bóng xanh trên bảng sau khi người thứ nhất chọn hình vuông có góc trái trên
// (x, y) và người thứ hai chọn hình vuông có góc trái trên (i, j)
int getNumBlue(int x, int y, int i, int j) {
    int res = cntBlue;
    res += getSum(x, x + sz - 1, y, y + sz - 1); // tổng điểm trong vùng 1
    res += getSum(i, i + sz - 1, j, j + sz - 1); // tổng điểm trong vùng 2

    // nếu hai vùng có giao nhau, trừ đi 2 lần tổng trong phần giao
    // vùng 1 có các hàng x -> x + sz - 1, vùng 2 có các hàng i -> i + sz - 1
    // phần giao có các hàng từ max(x, i) -> min(x, i) + sz - 1
    // tương tự với các cột
    res -= 2 * getSum(max(x, i), min(x, i) + sz - 1, max(y, j), min(y, j) + sz - 1);

    return res;
}

```



```

// trường hợp tệ nhất nếu người thứ nhất chọn hình vuông có góc trái trên là (x, y).

```

```

int calcResult(int x, int y) {
    int res = INF;
    // xét tất cả các cách đi của người thứ hai:
    for (int i = 1; i <= numRows - sz + 1; i++) for (int j = 1; j <= numCol - sz + 1; j++)
        minimize(res, getNumBlue(x, y, i, j));
    return res;
}

// xét các nước đi của người chơi thứ nhất và tính kết quả
for (int x = 1; x <= numRows - sz + 1; x++) {
    for (int y = 1; y <= numCol - sz + 1; y++) printf("%d ", calcResult(x, y));
    printf("\n");
}

```

Subtask 3

Ở hai subtask trên, ta chưa sử dụng ràng buộc $p \leq 20$ và giải như với p bình thường. Trong subtask cuối, với giới hạn $m, n \leq 500$ ta không thể duyệt toàn bộ tất cả các bộ (x, y, i, j) được. Do đó, ta cần cách xử lý thông minh hơn.

Ta vẫn phải xét toàn bộ các (x, y) vì đề bài yêu cầu với mỗi (x, y) tìm ra số bi xanh nhỏ nhất. Tuy nhiên, khi tính kết quả cho một nước chơi (x, y) ($\text{muy}(x, y)$); ta chia ra làm hai trường hợp:

- + Khi vùng (i, j) có phần chung với vùng (x, y)
- + Khi vùng (i, j) không có phần chung với vùng (x, y) .

Ta xét trường hợp thứ nhất: hai vùng chọn có phần chung.

Để hình vuông $p \times p$ màu đỏ có ít nhất một ô chung với hình vuông $p \times p$ màu xanh lam, góc trái trên của hình vuông màu đỏ phải nằm trong vùng màu xanh lá cây. Vùng màu xanh lá cây là một hình vuông kích thước $2p-1$ có góc trái trên là $(x - p + 1, y - p + 1)$.

Do ràng buộc $p \leq 20$, vùng màu xanh lá có mỗi chiều $\leq 40 \rightarrow$ số hình vuông màu đỏ có giao với hình xanh lam là không quá 40^2 . Do đó, ta có thể duyệt qua toàn bộ tất cả các hình vuông của trường hợp này và tính điểm như ở subtask 2.

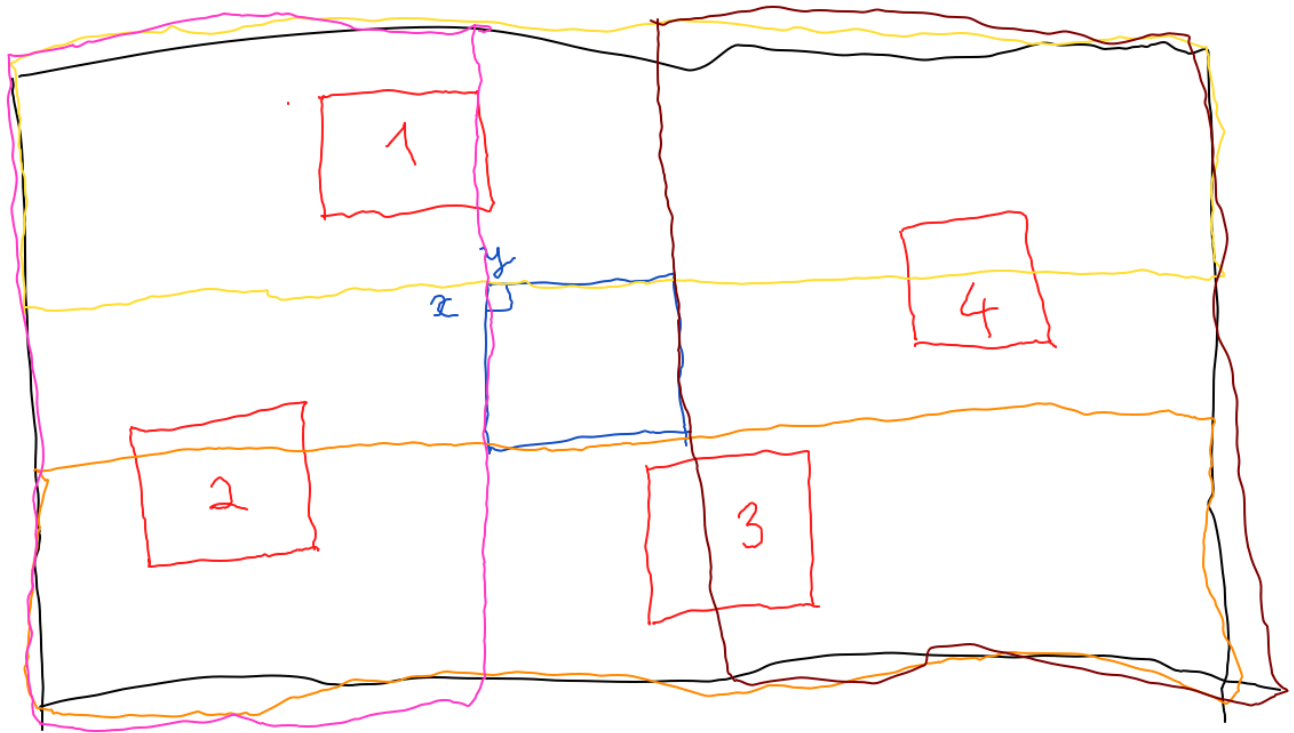
Giờ ta xét trường hợp hai hình không giao nhau. Trong trường hợp này, số bi xanh sau khi biến đổi bằng tổng của tất cả các ô trong hai vùng được chọn. Vì hai vùng là độc lập nhau, nên để tìm ra trường hợp tệ nhất (số bi xanh nhỏ nhất), ta cần chọn hình vuông của người chơi thứ hai sao cho tổng số điểm bên trong là nhỏ nhất.

Vì vậy, ta cần giải bài toán sau: Tìm hình vuông $p \times p$ có tổng số điểm (tổng giá trị các ô bên trong) là nhỏ nhất sao cho hình này không có phần chung với hình vuông (x, y) .

Nhận xét: Hình vuông $p \times p$ có góc trái trên là x, y phủ các hàng từ x đến $x + p - 1$ và các cột từ y đến $y + p - 1$. Do đó, ta có 4 khu vực như sau:

- + KV1: Toàn bộ các hàng phía trên hình vuông: $1 \rightarrow x - 1$
- + KV2: Toàn bộ các hàng phía dưới hình vuông: $x + p \rightarrow m$
- + KV3: Tất cả các cột bên trái hình vuông: $1 \rightarrow y - 1$
- + KV4: Tất cả các cột bên phải hình vuông: $y + p \rightarrow n$

Để một hình không có phần chung với hình vuông $p \times p$ có góc trái trên là (x, y) ; hình này phải nằm trọn vẹn ở một trong 4 khu vực kể trên.



Do đó, để giải quyết bài toán “tìm hình vuông $p \times p$ có tổng giá trị nhỏ nhất không giao với hình vuông của người chơi thứ nhất (có góc trái trên (x, y) ”, ta chia làm 4 bài toán tìm hình vuông tổng nhỏ nhất ở 4 khu vực nêu trên. Chú ý rằng, ta cần giải bài toán trên với mọi cặp (x, y) chứ không chỉ riêng một cặp.

Vì vậy, ta cần tính các mảng như sau:

- + $\text{bestTop}[x]$ = hình vuông $p \times p$ có tổng giá trị nhỏ nhất nằm trọn trong các hàng từ 1 đến x
- + $\text{bestBot}[x]$ = hình vuông $p \times p$ có tổng giá trị nhỏ nhất nằm trọn trong các hàng từ x đến m
- + $\text{bestLeft}[y]$ = hình vuông $p \times p$ có tổng giá trị nhỏ nhất nằm trọn trong các cột từ 1 đến y
- + $\text{bestRight}[y]$ = hình vuông $p \times p$ có tổng giá trị nhỏ nhất nằm trọn trong các cột từ y đến n .

Cách tính 4 mảng trên như sau: Chú ý, ở phần này, ta vẫn quy đổi các bi đỏ là +1 và các bi xanh là -1 như ở subtask 2. Sau đó, ta cũng cần tính mảng tổng dồn và hàm `getSum` để tính tổng một hình chữ nhật con trong $O(1)$.

```
int bestTop[MAX], bestBot[MAX], bestLeft[MAX], bestRight[MAX];
// khởi tạo các mảng best với giá trị dương vô cùng
memset(bestTop, 0x3f, sizeof bestTop);
memset(bestBot, 0x3f, sizeof bestBot);
memset(bestLeft, 0x3f, sizeof bestLeft);
memset(bestRight, 0x3f, sizeof bestRight);
```

```
for (int i = 1; i <= numRows - sz + 1; i++) for (int j = 1; j <= numCol - sz + 1; j++)
    // xét hình vuông  $p \times p$  với góc trái trên là  $(i, j)$ :
    // phủ các hàng  $i \rightarrow i + p - 1$  và phủ các cột từ  $j$  đến  $j + p - 1$ 
    int sum = getSum(i, i + sz - 1, j, j + sz - 1);
    minimize(bestTop[i + sz - 1], sum);
    minimize(bestBot[i], sum);
    minimize(bestLeft[j + sz - 1], sum);
    minimize(bestRight[j], sum);
```

```

// tới thời điểm này, bestTop[x] = giá trị nhỏ nhất của hình vuông có  $i + p - 1 = x$ 
// bằng việc áp dụng vòng lặp ở dưới, ta có bestTop[x] = giá trị nhỏ nhất của hình vuông có
 $i + p - 1 \leq x$ 
// tối ưu dần
for (int i = 1; i < numRows; i++) minimize(bestTop[i + 1], bestTop[i]);
for (int i = numRows; i > 1; i--) minimize(bestBot[i - 1], bestBot[i]);
for (int j = 1; j < numCol; j++) minimize(bestLeft[j + 1], bestLeft[j]);
for (int j = numCol; j > 1; j--) minimize(bestRight[j - 1], bestRight[j]);

```

Sau khi đã có mảng best như trên, ta viết hàm calcResult(x, y) - tính số bi xanh tối thiểu nếu người chơi thứ nhất chọn vùng có góc trái trên là (x, y) như sau:

```

// trường hợp tệ nhất nếu người thứ nhất chọn hình vuông có góc trái trên là (x, y).
int calcResult(int x, int y) {
    int res = INF;
    // trường hợp 1: vùng chọn của hai người chơi có giao nhau -> xét mọi khả năng
    // để hình vuông (x, y) giao với hình vuông (i, j)
    // chú ý không xét các hình vuông bị tràn ra ngoài bảng
    for (int i = max(1, x - sz + 1); i <= min(x + sz - 1, numRows - sz + 1); i++)
        for (int j = max(1, y - sz + 1); j <= min(y + sz - 1, numCol - sz + 1); j++)
            minimize(res, getNumBlue(x, y, i, j));

    // trường hợp 2: vùng chọn của hai người không giao nhau -> dùng mảng best
    // của 4 khu vực để tìm hình vuông nhỏ nhất
    int tmp = cntBlue + getSum(x, x + sz - 1, y, y + sz - 1);
    minimize(res, tmp + bestTop[x - 1]);
    minimize(res, tmp + bestBot[x + sz]);
    minimize(res, tmp + bestLeft[y - 1]);
    minimize(res, tmp + bestRight[y + sz]);

    return res;
}

```

Giải thích cách tính 4 mảng bestTop, bestBot, bestLeft, bestRight

Bài toán phụ

Ta xét bài toán như sau: Cho một mảng $a[1], a[2], \dots, a[n]$. Ban đầu, giá trị của tất cả các phần tử này là dương vô cùng. Ta cần thực hiện q thao tác, trong mỗi thao tác có hai số p và v ; ta cần gán $a[p] = \min(a[p], v)$. Sau khi thực hiện hết q thao tác, ta cần tính mảng $b[1], b[2], \dots, b[n]$ trong đó $b[i] = \min(a[1], a[2], \dots, a[i])$. In ra giá trị mảng b .

Ví dụ:

Input

```

7 9 // mảng n = 7 phần tử, có q = 9 thao tác
1 10 // a[1] = min(a[1], 10)
2 5  // a[2] = min(a[2], 5)
6 9  // a[6] = min(a[6], 9)
3 8  // a[3] = min(a[3], 8)

```

```

5 1
2 20 // a[2] = min(a[2], 20) - không có tác dụng do trước đó a[2] = 5
3 6
7 2
5 3

```

Output

```
10 5 5 5 1 1 1
```

Giải thích

	1	2	3	4	5	6	7
Mảng A:	10	5	6	INF	1	9	INF
Mảng B:	10	5	5	5	1	1	1

Chú ý: ta chỉ cần in ra mảng B sau khi đã thực hiện toàn bộ tất cả q thao tác

Cách làm bài toán: Mỗi truy vấn có hai số p và v. Yêu cầu của truy vấn này là gán $a[p] = \min(a[p], v)$. Mặt khác, trong dãy cần in ra, số thứ i là min của i số đầu tiên trong mảng a. Vì vậy, ta có $b[i] = \min(a[1], a[2], \dots, a[i]) = \min(b[i - 1], a[i])$

Nhận xét: Nếu như ta đã quen với mảng tổng tiền tố (hay còn gọi là mảng tổng dồn), ta thấy mảng b giống như mảng “min tiền tố”. Nói cách khác, mảng b có cách hoạt động giống với mảng tổng tiền tố, nhưng thay phép + bằng phép min. Vì vậy, ở công thức truy hồi, ta có $b[i] = \min(b[i - 1], a[i])$ thay vì $sum[i] = sum[i - 1] + a[i]$ đối với mảng tổng tiền tố thông thường.

Với bài toán phụ, ta có một cách để chỉ dùng một mảng thay vì cần phải khai báo cả hai mảng a và b.

```

int n, q, a[MAX]; cin >> n >> q;
// khởi tạo mảng bằng dương vô cùng
for (int i = 1; i <= n; i++) a[i] = INF;

// thực hiện lần lượt q thao tác
for (int i = 1; i <= q; i++) {
    int p, v; cin >> p >> v;
    a[p] = min(a[p], v);
}

// vòng lặp dưới đây chuyển từ mảng a sang mảng b
// nói cách khác, sau vòng lặp dưới đây, phần tử a[i]
// mang giá trị min(a[1], a[2], ..., a[i]) bước trước
for (int i = 2; i <= n; i++) minimize(a[i], a[i - 1]);

// in kết quả:
for (int i = 1; i <= n; i++) cout << a[i] << " "; cout << "\n";

```

Trong đoạn code màu xanh lá cây, ban đầu từ mảng a ta đã biến thành mảng b. Nói cách khác, sau khi thực hiện vòng lặp màu xanh lá cây, giá trị của $a[i]$ trở thành $\min(a[1], a[2], \dots, a[i])$.

Ví dụ, nếu trước vòng lặp, giá trị của mảng a là 10 5 6 INF 1 9 INF thì sau vòng lặp trở thành

10 5 5 5 1 1 1

Phân tích:

- + Khi $i = 2$, ta gán $a[2] = \min(a[2], a[1]) = \min(10, 5) = 5$
- + Khi $i = 3$, ta gán $a[3] = \min(a[3], a[2]) = \min(5, 6) = 5$
- + Khi $i = 4$, ta gán $a[4] = \min(a[4], a[3]) = \min(\text{INF}, 5) = 5$
- + Khi $i = 5$, ta gán $a[5] = \min(a[5], a[4]) = \min(1, 5) = 1$

Trở lại với việc tính 4 mảng best

Giờ ta phân tích cách tính mảng bestTop

Nhắc lại yêu cầu: Cho một bảng $m \times n$ (m hàng và n cột), ta cần dựng mảng bestTop[1...m] với ý nghĩa: Với mọi x từ 1 đến m , bestTop[x] = tổng giá trị nhỏ nhất của một hình vuông $p \times p$ nằm trọn trong vùng x hàng đầu tiên (các hàng từ 1 đến x)

Để một hình vuông nằm trọn trong vùng x hàng đầu tiên, cạnh dưới cùng của hình vuông này phải $\leq x$. Một hình vuông $p \times p$ được đại diện bởi tọa độ góc trái trên (i, j) . Khi đó hình vuông sẽ chứa các hàng từ i đến $i+p-1$ và chứa các cột từ j đến $j+p-1$. Để hình vuông có góc trái trên (i, j) nằm trọn trong x hàng đầu tiên, ta phải có cạnh dưới cùng $i+p-1 \leq x$.

Như vậy, mảng bestTop[x] có thể được hiểu là hình vuông có giá trị nhỏ nhất sao cho $i+p-1 \leq x$. Vì thế, ta tính mảng bestTop[x] này như sau: đầu tiên ta duyệt qua tất cả các hình vuông (i, j) hợp lệ. Với mỗi hình vuông như vậy, ta gán $\text{bestTop}[i+p-1] = \min(\text{bestTop}[i+p-1], \text{giá trị của hình vuông})$. Sau khi đã liệt kê tất cả các hình vuông hợp lệ, ta xét lần lượt các i từ 2 đến m , ta gán $\text{bestTop}[i] = \min(\text{bestTop}[i], \text{bestTop}[i-1])$.

Với mảng bestBot, ta thực hiện theo cách tương tự: $\text{bestBot}[x]$ = giá trị nhỏ nhất của hình vuông $p \times p$ nằm trọn trong vùng các hàng từ x tới m . Để một hình vuông có góc trái trên là (i, j) nằm trọn trong vùng từ hàng x tới hàng m , ta phải có $i \geq x$. Do đó, $\text{bestBot}[x]$ được hiểu là giá trị nhỏ nhất của hình vuông $p \times p$ có $i \geq x$.

Tương tự như việc tính bestTop, đầu tiên ta cần phải liệt kê tất cả các hình vuông $p \times p$ có thể, với mỗi hình vuông như thế, ta gán $\text{bestBot}[i] = \min(\text{bestBot}[i], \text{giá trị của hình vuông})$. Như vậy, sau bước liệt kê mọi hình vuông, ta đang có $\text{bestBot}[x]$ = hình vuông có $i = x$ (cạnh trên = x).

Để biến từ ' $\text{bestBot}[x]$ = hình vuông có $i = x$ ' sang ' $\text{bestBot}[x]$ = hình vuông có $i \geq x$ ' ta cũng sử dụng kỹ thuật mảng min dồn giống như ở trên. Tuy nhiên, do ở đây $\text{bestBot}[x]$ là ngược với bestTop ($\text{bestTop}[x]$ chứa những cái từ 1 đến x , hay $\leq x$), $\text{bestBot}[x]$ chứa những cái $\geq x$ hay từ x đến m , do đó, mảng $\text{bestBot}[x]$ dồn từ dưới lên (giảm dần) thay vì trên xuống (Tăng dần)