

2023 summer contest #29, university training, round A4

Statement
is not
available
on
English
language

A. Nàng Bạch Tuyết và bảy chú lùn

0.25 seconds, 8 megabytes

Một ngày nọ, nàng Bạch Tuyết cùng bảy chú lùn đi dạo chơi ở công viên.

Vì mãi chơi và mãi thưởng thức cảnh đẹp xung quanh, nàng Bạch Tuyết bị lạc và không thể tìm thấy bảy chú lùn. Vì công viên rất rộng, lại đông người ghé chơi, việc tìm thấy những chú lùn đang lẫn khuất đâu đây là không hề đơn giản. Vì vậy, nàng Bạch Tuyết quyết định khóa tất cả các cửa ngõ ra khỏi công viên, và đi dò tìm mọi chàng trai đang ở đây.

Lúc đó, b chàng trai đang ở trong công viên. Toàn bộ b người này đã nói tên và chiều cao của mình cho nàng Bạch Tuyết. Dù sống chung với bảy chú lùn đã lâu, Bạch Tuyết lại chưa bao giờ hỏi tên các chú lùn. Bạch Tuyết chỉ nhớ rằng, cả bảy chú lùn đều có chiều cao bằng nhau, và không ai khác cao bằng những chú lùn này.

Do số lượng chàng trai rất lớn, nàng Bạch Tuyết nhờ bạn viết chương trình, đọc vào tên và chiều cao của b chàng trai trong công viên, in ra tên của bảy chú lùn.

Input

Dòng đầu tiên chứa số nguyên b ($7 \leq b \leq 100$) — số chàng trai trong công viên.

Trong b dòng tiếp theo, dòng thứ i chứa xâu kí tự n_i và số nguyên h_i . Trong đó, n_i là tên của chàng trai thứ i , chứa từ 1 đến 10 chữ cái tiếng Anh, còn số h_i ($1 \leq h_i \leq 1000$) là chiều cao của chàng trai này.

Output

In ra bảy xâu kí tự lần lượt là tên của bảy chú lùn, được sắp xếp theo thứ tự từ điển tăng dần. Dữ liệu vào đảm bảo có thể xác định duy nhất bảy chú lùn từ b chàng trai, theo quy tắc ở trên.

input
9 city 1 of 1 stars 1 are 2 you 1 shining 1 just 2 for 1 me 1
output
city for me of shining stars you

input
10 vui 7 la 8 con 9 gai 10 ha 7 la 7 con 7 gai 7 that 7 tuyet 7
output
con gai ha la that tuyet vui

input
7 gs 227 pvh 227 cute 227 a 227 hihi 227 do 227 ngoc 227
output
a cute do gs hihi ngoc pvh

Statement
is not
available
on
English
language

B. Đồ thị tam giác

0.25 seconds, 128 megabytes

Một đồ thị vô hướng được gọi là "đồ thị tam giác" khi và chỉ khi tồn tại ba đỉnh phân biệt u , v và w sao cho ba đỉnh này đôi một được nối trực tiếp với nhau.

Cho một đồ thị vô hướng gồm n đỉnh và m cạnh. Các đỉnh được đánh số từ 1 đến n . Hãy thêm ít cạnh nhất vào đồ thị sao cho đồ thị này là "đồ thị tam giác".

Input

Dòng đầu tiên chứa hai số nguyên n và m ($3 \leq n \leq 50$, $0 \leq m \leq 10^9$), lần lượt là số đỉnh và số cạnh của đồ thị.

Trong m dòng còn lại, mỗi dòng chứa hai số nguyên u và v ($1 \leq u, v \leq n$) cho biết hai đỉnh u và v của đồ thị được nối trực tiếp bởi một cạnh. Dữ liệu vào đảm bảo đồ thị được cho là đơn đồ thị; tức hai đỉnh bất kì chỉ được nối trực tiếp với nhau bởi không quá một cạnh, và không cạnh nào nối một đỉnh với chính nó.

Output

In ra một số nguyên duy nhất là số cạnh tối thiểu cần thêm để đồ thị đã cho là đồ thị tam giác.

input
4 4 1 4 3 1 2 3 3 4
output
0

input
6 3 1 4 2 1 3 5
output
1

Trong ví dụ thứ nhất, đồ thị đã có ba cạnh (1, 3), (3, 4) và (4, 1) nên đã là đồ thị tam giác mà không cần thêm cạnh nào.

Trong ví dụ thứ hai, ta cần thêm cạnh (2, 4); để kết hợp với các cạnh có sẵn (1, 4) và (2, 1) ta được một đồ thị tam giác.

Statement
is not
available
on
English
language

C. Đồ thị liên thông mạnh

2 seconds, 256 megabytes

Đồ thị liên thông mạnh là đồ thị có hướng thỏa mãn tính chất sau: Với mọi cặp đỉnh u và v bất kì, tồn tại đường đi từ u đến v .

Cho một đồ thị có hướng có trọng số, hãy chọn ra một số cạnh của đồ thị này sao cho tập hợp các cạnh được chọn cùng với n đỉnh của đồ thị gốc tạo ra một đồ thị liên thông mạnh. Trong các cách chọn cạnh, hãy tìm cách chọn sao cho chênh lệch giữa trọng số lớn nhất và trọng số nhỏ nhất của các cạnh được chọn là nhỏ nhất.

Input

Dòng đầu tiên chứa hai số nguyên n và m ($1 \leq n \leq 3000, 1 \leq m \leq 5000$), lần lượt là số đỉnh và số cạnh của đồ thị.

Trong m dòng còn lại, dòng thứ i chứa ba số nguyên u_i, v_i và c_i ($1 \leq u_i, v_i \leq n, 0 \leq c_i \leq 10^9$), cho biết cạnh thứ i của đồ thị xuất phát từ đỉnh

Output

In ra một số nguyên duy nhất là giá trị nhỏ nhất của chênh lệch giữa trọng số lớn nhất và trọng số nhỏ nhất của các cạnh được chọn, hoặc -1 nếu không tồn tại cách chọn thỏa mãn.

Scoring

- Subtask 1 (12 điểm): $n \leq 30$ và $m \leq 50$
- Subtask 2 (16 điểm): $n \leq 90$ và $m \leq 150$
- Subtask 3 (20 điểm): $n \leq 270$ và $m \leq 450$
- Subtask 4 (24 điểm): $n \leq 1080$ và $m \leq 1800$
- Subtask 5 (28 điểm): $n \leq 3000$ và $m \leq 5000$

input
3 6 1 2 8 2 3 32 3 1 16 1 3 81 3 2 243 2 1 27
output
24

Statement
is not
available
on
English
language

D. Đồ thị ước số chung lớn nhất

2 seconds, 512 megabytes

Cho một dãy gồm n số nguyên a_1, a_2, \dots, a_n và một số nguyên k . Ta dựng một đồ thị vô hướng gồm n đỉnh, các đỉnh được đánh số từ 1 đến n . Giữa hai đỉnh i và j có cạnh nối trực tiếp khi và chỉ khi ước số chung lớn nhất của a_i và a_j lớn hơn k .

Bạn được cho q cặp đỉnh $(x_1, y_1), (x_2, y_2), \dots, (x_q, y_q)$. Với cặp đỉnh thứ t , bạn cần trả lời câu hỏi: trên đồ thị có tồn tại đường đi giữa hai đỉnh x_t và y_t hay không.

Input

Dòng đầu tiên chứa ba số nguyên n, k và q ($1 \leq n, q \leq 10^6, 1 \leq k \leq 10^7$).

Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^7$).

Trong q dòng cuối cùng, dòng thứ t chứa hai số nguyên x_t và y_t ($1 \leq x_t, y_t \leq n$).

Output

In ra một xâu gồm q kí tự; kí tự thứ t là Y nếu tồn tại đường đi giữa hai đỉnh x_t và y_t , là N nếu không tồn tại người đi như vậy.

Scoring

- Subtask 1 (29 điểm): $n \leq 300$ và $q \leq 300$
- Subtask 2 (34 điểm): $n \leq 300$ và $q \leq 10^6$
- Subtask 3 (37 điểm): $n \leq 10^6$ và $q \leq 10^6$

input
3 2 1 8 12 9 1 3
output
Y

input
5 1 2 2 4 6 8 10 3 4 3 5
output
YY

Trong m dòng còn lại, mỗi dòng chứa hai số nguyên u và v ($1 \leq u, v \leq n, \sqrt{\frac{u^2+v^2}{2}} > \frac{2}{\frac{1}{u}+\frac{1}{v}}$), cho biết đồ thị có một cạnh nối hai đỉnh u và v . Dữ liệu vào đảm bảo đồ thị được cho là đơn đồ thị; tức hai đỉnh bất kì chỉ được nối trực tiếp với nhau bởi không quá một cạnh, và không cạnh nào nối một đỉnh với chính nó.

Output

In ra một số nguyên duy nhất là số bộ đỉnh và cạnh của đồ thị tạo ra đồ thị hình người.

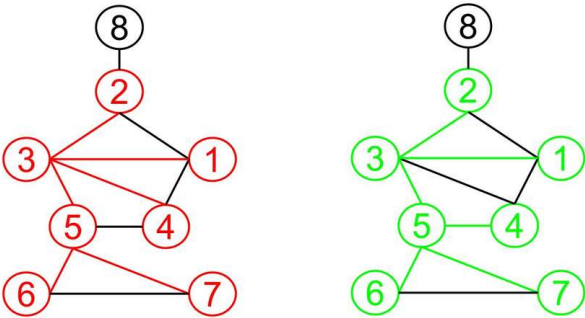
Scoring

- Subtask 1 (18 điểm): $n \leq 20$
- Subtask 2 (18 điểm): $n \leq 200$
- Subtask 3 (20 điểm): $n \leq 2000$ và $m \leq 10000$
- Subtask 4 (22 điểm): $n \leq 20000$ và $m \leq 100000$
- Subtask 5 (22 điểm): $n \leq 200000$

input
8 11 1 2 2 3 3 1 1 4 4 5 3 4 3 5 5 6 6 7 7 5 2 8
output
2

input
7 8 1 2 1 3 1 4 1 5 1 6 5 6 7 5 7 6
output
2

Hai hình vẽ dưới đây mô tả hai cách tạo ra "đồ thị hình người" từ đồ thị đã cho trong ví dụ đầu tiên. Các đỉnh và các cạnh được tô màu là các đỉnh và các cạnh được chọn.



input
4 2 4 22 7 19 97 1 2 2 3 3 4 4 4
output
NNNY

Trong ví dụ thứ nhất, đồ thị có các cạnh $(1, 2)$ và $(2, 3)$ do $\gcd(8, 12) = 4 > 2$ và $\gcd(12, 9) = 3 > 2$. Do đó, có một đường đi từ đỉnh 8 tới đỉnh 9: $8 \rightarrow 12 \rightarrow 9$.

Trong ví dụ thứ hai, đồ thị đã cho là đồ thị đầy đủ.

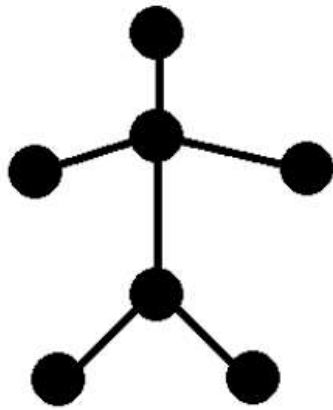
Trong ví dụ thứ ba, đồ thị không có cạnh nào.

Statement is not available on English language

E. Đồ thị hình người

0.5 seconds, 256 megabytes

Đồ thị hình người là một đồ thị vô hướng gồm 7 đỉnh và 6 cạnh, sao cho có thể vẽ thành dạng người như hình dưới đây:



Cho một đồ thị vô hướng với n đỉnh và m cạnh; các đỉnh được đánh số từ 1 đến n . Hãy đếm số cách chọn ra từ đồ thị 7 đỉnh và 6 cạnh nối giữa các đỉnh này sao cho các đỉnh và cạnh được chọn tạo thành một đồ thị hình người như ở trên.

Hai cách chọn được coi là khác nhau khi và chỉ khi tập đỉnh được chọn hoặc tập cạnh được chọn trong hai cách là khác nhau.

Do kết quả có thể rất lớn, bạn chỉ cần in ra phần dư của số cách khi chia cho 998244353.

Input

Dòng đầu tiên chứa hai số nguyên n và m ($7 \leq n \leq 2 \cdot 10^5, 6 \leq m \leq 3 \cdot 10^5$), lần lượt là số đỉnh và số cạnh của đồ thị.

Statement is not

available
on
English
language

F. Đường tới trường

1 second, 256 megabytes

Năm nay bé Lin vào cấp ba. Ở tuổi 15, là một cô bé mới lớn với những rung động đầu đời, bé không muốn được mẹ dắt tay đến trường nữa. Bé muốn đi một mình để có thể vừa đi vừa hát, vừa khám phá thế giới thú vị quanh bé. Bé muốn hòa mình vào không gian bao la, ngắm mây xanh và nắng đẹp. Bé muốn tranh thủ vui đùa và thả thính những chàng trai lạ mà mình gặp ngoài đường. Và bé Lin còn thích nhiều thứ nữa...

Thế nhưng, cuộc đời không đơn giản như vậy. Để có thể tới trường với mọi thứ còn nguyên vẹn, bé phải nhớ đường đi để chắc rằng mình không đi về một nơi xa xăm kì bí, không bị soái ca nào làm cho lạc lối, hay không phiêu du nơi miền cực lạc. Và bây giờ là lúc để Lin học thuộc bản đồ khu vực quanh nhà và trường.

Nơi Lin sinh sống có thể được minh họa bởi một lưới ô vuông gồm r hàng và c cột với các hàng được đánh số từ 1 đến r và các cột được đánh số từ 1 đến c . Ô nằm ở vị trí giao của hàng i và cột j được kí hiệu là (i, j) . Một số ô trên lưới là nhà hàng xóm, siêu thị hoặc các công trình khác mà Lin không thể đi vào. Các ô còn lại là đường xá, nhà Lin và trường học — những nơi Lin có thể đến hoặc đi qua. Nhà Lin nằm ở ô (h_x, h_y) , còn trường Lin đang theo học nằm ở ô (s_x, s_y) . Trong mỗi bước, Lin chỉ được đi tới một ô có chung cạnh với vị trí hiện tại.

Dù bé nhưng Lin khôn như trâu, sẵn sàng đi bộ cả một chặng dài. Lin còn thích điều này vì đây là cơ hội để Lin thưởng thức nhiều hoa thơm cỏ lạ. Thế nhưng trí nhớ của Lin không tốt nên không thể nhớ được những con đường ngoằn ngoèo với nhiều khúc rẽ. Các bạn hãy giúp Lin tìm ra một con đường từ nhà tới trường với ít lần chuyển hướng nhất có thể nhé. Chỉ có điều, Lin sẽ không chịu đi nếu phải chuyển hướng nhiều hơn k lần, vì vậy các bạn phải báo trước điều này cho Lin không thì Lin sẽ đi khóc nhè đó.

Để giảm kích thước file dữ liệu đầu vào, lưới ô vuông thể hiện khu vực Lin sinh sống không được cho trực tiếp (dưới dạng r xâu kí tự độ dài c). Thay vào đó, mỗi hàng của lưới sẽ được nén bởi một xâu gồm $\lceil \frac{c}{6} \rceil$ ký tự. Các ký tự này có thể là chữ số (0 - 9), chữ cái in hoa (A - Z), chữ cái in thường (a - z) hay các kí tự đặc biệt # và \$. Cách giải nén để có được một hàng như sau:

- Đầu tiên, ta quy đổi các kí tự trong xâu được cho thành các số nguyên từ 0 đến 63 theo quy tắc:
 - Các chữ số từ 0 đến 9 lần lượt tương ứng với các số từ 0 đến 9.
 - Các chữ cái in hoa từ A đến Z lần lượt tương ứng với các số từ 10 đến 35.
 - Các chữ cái in thường từ a đến z lần lượt tương ứng với các số từ 36 đến 61.
 - Ký tự # tương ứng với số 62.
 - Ký tự \$ tương ứng với số 63.
- Ta biểu diễn mỗi số sau quy đổi bằng một xâu nhị phân độ dài 6 bằng cách phân tích số đó ra hệ nhị phân, viết các chữ số theo thứ tự đảo ngược: bit đơn vị nằm ở ngoài cùng bên trái, bit tiếp theo đến bit hàng chục, ..., bit lớn nhất nằm ở ngoài cùng bên phải.
- Ghép các xâu nhị phân độ dài 6 vừa tìm được thành một dãy dài, lấy ra c kí tự đầu tiên. Đây chính là c kí tự thể hiện một hàng của lưới ô

Problems - Codeforces

vuông, trong đó kí tự 0 đại diện cho một ô mà Lin có thể đi vào, kí tự 1 đại diện cho một ô mà Lin không thể đi vào.

Xem phần giải thích ví dụ để hiểu rõ hơn về cách xây dựng lưới ô vuông.

Input

Dòng đầu tiên chứa ba số nguyên r , c và k ($1 \leq r, c \leq 2000$, $0 \leq k \leq 10^7$) thể hiện kích thước của lưới ô vuông minh họa khu vực Lin sinh sống và số lần rẽ nhiều nhất mà Lin có thể nhớ.

Dòng thứ hai chứa bốn số nguyên h_x, h_y, s_x, s_y ($1 \leq h_x, s_x \leq r$, $1 \leq h_y, s_y \leq c$) thể hiện vị trí của nhà và trường của Lin. Dữ liệu vào đảm bảo đây là hai ô vuông khác nhau trong lưới mà Lin có thể đi vào.

Trong r dòng cuối cùng, dòng thứ i chứa một xâu kí tự độ dài $\lceil \frac{c}{6} \rceil$ mô tả hàng thứ i của lưới ô vuông mô tả khu vực sinh sống của Lin theo quy tắc nêu ở trên.

Output

Nếu không tồn tại con đường từ nhà tới trường với tối đa k lần rẽ, in ra một số nguyên - 1 duy nhất. Ngược lại, in ra đáp án theo khuôn dạng sau:

- Dòng đầu tiên chứa số nguyên t ($0 \leq t \leq k$) là số lần rẽ tối thiểu Lin cần thực hiện để đi từ nhà đến trường.
- Trong $t + 2$ dòng còn lại, dòng thứ i ($0 \leq i \leq t$) chứa hai số nguyên x_i và y_i ($1 \leq x_i \leq r$, $1 \leq y_i \leq c$) mô tả con đường từ nhà đến trường với ý nghĩa: Lin cần đi qua các ô theo thứ tự $(x_0, y_0) \rightarrow (x_1, y_1) \rightarrow (x_2, y_2) \rightarrow \dots \rightarrow (x_t, y_t) \rightarrow (x_{t+1}, y_{t+1})$. Trong đó, việc đi từ ô (x_j, y_j) tới ô (x_{j+1}, y_{j+1}) (với mọi $0 \leq j \leq t$) là đi thẳng (tức không phải rẽ lần nào).

Một con đường hợp lệ cần thỏa mãn các điều kiện sau đây:

- $x_0 = h_x$ và $y_0 = h_y$.
- $x_{t+1} = s_x$ và $y_{t+1} = s_y$.
- Với mọi $0 \leq j \leq t$, $(x_j - x_{j+1}) \cdot (y_t - y_{t+1}) = 0$ và $(x_j - x_{j+1})^2 + (y_t - y_{t+1})^2 > 0$.

Nếu có nhiều con đường tối ưu, bạn được phép in ra con đường bất kì.

Scoring

- Subtask 1 (13 điểm): $k \leq 0$
- Subtask 2 (13 điểm): $k \leq 1$
- Subtask 3 (15 điểm): $k \leq 2$
- Subtask 4 (15 điểm): $m, n \leq 400$ và $k \leq 30$
- Subtask 5 (20 điểm): $m, n \leq 400$
- Subtask 6 (24 điểm): Không có ràng buộc gì thêm.

Với mỗi test, bạn được 50% số điểm nếu chỉ ra số lần rẽ tối thiểu nhưng không chỉ ra được một con đường tối ưu.

input
<pre>5 10 1 3 1 5 5 hy sG 08 At a\$</pre>
output
<pre>1 3 1 3 5 5 5</pre>

input
11 20 3 3 14 9 3 F19M 919c FYuJ 1K8g 1886 000m FfpJ 19H4 19n3 19H0 FFnx
output
3 3 14 6 14 6 5 9 5 9 3

input
11 20 2 3 14 9 3 F19M 919c FYuJ 1K8g 1886 000m FfpJ 19H4 19n3 19H0 FFnx
output
-1

input
7 6 2 1 1 7 1 0 R 0 U 0 R 0
output
2 1 1 1 6 7 6 7 1

Trong ví dụ thứ nhất, ta tiến hành khôi phục lại lưới ô vuông như sau:

- Đầu tiên, ta biến đổi các kí tự trong xâu nén thành các số nguyên từ 0 đến 63:
- 43 60

54 16

0 8

10 55

36 63
- Sau đó, ta biểu diễn mỗi số ở trên bằng một xâu nhị phân độ dài 6, chú ý rằng bit đơn vị được viết ngoài cùng bên trái và bit có giá trị lớn

nhất được viết ngoài cùng bên phải:

- $43 = 2^5 + 2^3 + 2^1 + 2^0$ có xâu biểu diễn là 110101.
 - $60 = 2^5 + 2^4 + 2^3 + 2^2$ có xâu biểu diễn là 001111.
 - $54 = 2^5 + 2^4 + 2^2 + 2^1$ có xâu biểu diễn là 011011.
 - $16 = 2^4$ có xâu biểu diễn là 000010
 - 0 có xâu biểu diễn là 000000.
 - $8 = 2^3$ có xâu biểu diễn là 000100.
 - $10 = 2^3 + 2^1$ có xâu biểu diễn là 010100.
 - $55 = 2^5 + 2^4 + 2^2 + 2^1 + 2^0$ có xâu biểu diễn là 111011.
 - $36 = 2^5 + 2^2$ có xâu biểu diễn là 001001.
 - $63 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ có xâu biểu diễn là 111111.
- Tiếp theo, ta viết các xâu nhị phân trên vào bảng:

110101 001111

011011 000010

000000 000100

010100 111011

001001 111111

- Cuối cùng, ta lấy ra 10 chữ số đầu tiên ở mỗi hàng và được lưới ô vuông như ở dưới đây. Trong đó, các kí tự # đại diện cho các ô Lin không thể đi qua, kí tự H đại diện cho nhà của Lin, kí tự S đại diện cho trường của Lin và các kí tự . đại diện cho các ô Lin có thể đi qua khác.

##.#.#.##

.##.##....

H.....#

.#.#.###.

..#.S#####

Với lưới ô vuông như trên, đường đi từ nhà đến trường cần ít nhất một lần rẽ, ví dụ như sau:

##.#.#.##

.##.##....

H>>>v....#

.#.#v.###.

..#.S#####

Trong ví dụ thứ hai, lưới ô vuông có dạng như sau:

####.#.....#.#...#

#..#.#.....#.#...#

####.#...#.H.#####

#.....#.#.....#

#.....#.....#...#

.....#.....#

####.#.#.###.####

#.....#.#.#.....#...

#.S...#.#.#.....####

#.....#.#.#.....#...

####.####.#.....####

Một trong những đường đi cần ít lần rẽ nhất là:

####.#.....#.#...#

```
#.#.#.#.#.#.#.#.#
####.#.#.#.#####
#.#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#
....V<<<<<<<<....
####V.#.#.#.####
#.#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#
####.#.#.#.#####
```

Statement
is not
available
on
English
language

G. SBB CFF FFS

0.75 seconds, 64 megabytes

Đất nước Thụy Sĩ nổi tiếng với những ngọn núi cao ngất và những hồ nước xanh trong vắt. Nơi đây không có những thành phố sầm uất như London, Paris hay những công trình nổi tiếng để thu hút khách du lịch. Thụy Sĩ lôi cuốn bởi vẻ đẹp hoang dã mà thiên nhiên ban tặng. Du khách năm châu tới đây chủ yếu để leo lên những đỉnh núi cao ngất như Rigi, Pilatus hay Alps; chiêm ngưỡng cảnh thiên nhiên hùng vĩ và thu trọn cả đất nước Thụy Sĩ vào tầm mắt.

Trong kỳ thực tập tại Google Zurich, GSPVH lên kế hoạch chinh phục hết các ngọn núi tại đây. Đất nước Thụy Sĩ có n ngọn núi, chia vào k khu vực. k khu vực này có thể giao nhau, bởi một ngọn núi có thể thuộc về nhiều hơn một khu vực. Nói cách khác, mỗi khu vực là một tập con của tập hợp gồm n ngọn núi. Để thuận tiện, các ngọn núi được đánh số từ 1 tới n , và ta coi như có $n + k$ khu vực, đánh số từ 1 tới $n + k$. Các khu vực từ 1 tới n chỉ có một ngọn núi (khu vực i chỉ chứa ngọn núi i), các khu vực từ $n + 1$ tới $n + k$ có ít nhất hai ngọn núi.

Trước khi lên kế hoạch khám phá n ngọn núi, GSPVH thu thập thông tin về độ hiểm trở của chúng thông qua những thực tập sinh khác. Theo đó, độ hiểm trở của mỗi ngọn núi thuộc một trong n mức, đánh số từ 1 tới n và không có hai ngọn núi nào có cùng độ hiểm trở. Ngoài ra, những người bạn của GSPVH còn cung cấp m mẫu thông tin, thuộc một trong bốn dạng sau:

- MAX $x \ y$ với $1 \leq y \leq n < x \leq n + k$, cho biết trong khu vực x , ngọn núi y có độ hiểm trở lớn nhất.
- MIN $x \ y$ với $1 \leq y \leq n < x \leq n + k$, cho biết trong khu vực x , ngọn núi y có độ hiểm trở nhỏ nhất.
- $x < y$ với $1 \leq x, y \leq n + k$, cho biết mọi ngọn núi ở khu vực x có độ hiểm trở không lớn hơn mọi ngọn núi ở khu vực y .
- $x > y$ với $1 \leq x, y \leq n + k$, cho biết mọi ngọn núi ở khu vực x có độ hiểm trở không nhỏ hơn mọi ngọn núi ở khu vực y .

GSPVH muốn chinh phục các đỉnh núi với độ hiểm trở tăng dần. Vì vậy các bạn hãy sắp xếp các ngọn núi theo thứ tự này nhé. Do thông tin thu thập được còn ít, có thể có nhiều thứ tự cùng thỏa mãn. Trong trường hợp đó, bạn nên đưa ra dãy có thứ tự từ điển nhỏ nhất. Dữ liệu vào đảm bảo có ít nhất một thứ tự hợp lệ.

Input

Problems - Codeforces

Dòng thứ nhất chứa ba số nguyên dương n, k và m ($\max(\frac{n}{2}, k, \frac{m}{3}) \leq 10^5$), lần lượt là số ngọn núi, số khu vực và số mẫu thông tin GSPVH thu được.

Trong k dòng tiếp theo, dòng thứ i gồm hai số nguyên x và y ($1 \leq x, y < n + i$) cho biết tập hợp các ngọn núi thuộc khu vực thứ $n + i$ là hợp của tập hợp các ngọn núi thuộc khu vực x và khu vực y .

Trong m dòng cuối cùng, mỗi dòng thể hiện một mẫu thông tin theo một trong bốn dạng nêu trên.

Output

Gồm một dòng duy nhất chứa n số, lần lượt là số thứ tự của các ngọn núi theo thứ tự độ hiểm trở tăng dần.

Scoring

Với mỗi test, bạn sẽ nhận được:

- 100% số điểm nếu đáp án của bạn thỏa mãn tất cả m mẫu thông tin và có thứ tự từ điển nhỏ nhất, hoặc
- 73% số điểm nếu đáp án của bạn thỏa mãn tất cả m mẫu thông tin, hoặc
- 41% số điểm nếu đáp án của bạn thỏa mãn tất cả các mẫu thông tin loại 3 và 4 ($x < y$ và $x > y$) với $x, y \leq n$, hoặc
- 17% số điểm nếu đáp án của bạn là hoán vị của các số tự nhiên từ 1 tới n , hoặc
- 0 điểm trong các trường hợp còn lại.

Ngoài ra, trong 20% số test, $\max(\frac{n}{2}, k, \frac{m}{3}) \leq 10^3$.

input
7 5 4 2 4 8 1 3 6 7 3 10 11 12 > 9 MIN 8 4 MAX 12 6 11 < 10
output
1 4 2 5 7 3 6

input
4 1 2 1 2 1 > 2 5 > 3
output
3 2 1 4

Trong ví dụ đầu tiên:

- Có $n = 7$ ngọn núi và $n + k = 12$ khu vực.
- Khu vực 1 có ngọn núi 1, khu vực 2 có ngọn núi 2, ..., khu vực 7 có ngọn núi 7.
- Khu vực 8 có các ngọn núi 2 và 4.
- Khu vực 9 có các ngọn núi 1, 2 và 4.
- Khu vực 10 có các ngọn núi 3 và 6.
- Khu vực 11 có các ngọn núi 7 và 3.
- Khu vực 12 có các ngọn núi 3, 6 và 7.

Trong ví dụ thứ hai:

- Nếu output của bạn là 4123, bạn được 17% số điểm của test.

- Nếu output của bạn là 4213, bạn được 41% số điểm của test.
- Nếu output của bạn là 4321, bạn được 73% số điểm của test.
- Nếu output của bạn là 3214, bạn được 100% số điểm của test.

Statement
is not
available
on
English
language

H. Khôi phục hoán vị

2.25 seconds, 256 megabytes

Một dãy số nguyên (p_1, p_2, \dots, p_n) được gọi là một hoán vị của các số tự nhiên từ 1 đến n khi và chỉ khi $1 \leq p_i \leq n$ với mọi $1 \leq i \leq n$, và các số p_1, p_2, \dots, p_n đôi một phân biệt. Từ hoán vị (p_1, p_2, \dots, p_n) , ta định nghĩa dãy số $(next_1, next_2, \dots, next_n)$ như sau: Với mọi $1 \leq i \leq n$, $next_i$ là chỉ số j nhỏ nhất sao cho $i < j \leq n$ và $p_j > p_i$. Nếu không có chỉ số j nào như vậy, ta coi $next_i = n + 1$.

Cho một dãy số nguyên (x_1, x_2, \dots, x_n) . Hãy tìm một dãy số nguyên (p_1, p_2, \dots, p_n) là hoán vị của các số nguyên từ 1 đến n sao cho với mọi $1 \leq i \leq n$, $next_i = x_i$ hoặc $x_i = -1$.

Input

Dòng đầu tiên chứa hai số nguyên θ và $\tau (1 \leq \theta \leq 5, 1 \leq \tau \leq 10^9)$, lần lượt là số thứ tự của subtask chứa test và số bộ dữ liệu có trong file input này.

Tiếp theo là τ bộ dữ liệu, mỗi bộ dữ liệu được mô tả theo khuôn dạng sau đây:

- Dòng thứ nhất chứa số nguyên $n (1 \leq n \leq 524288)$.
- Dòng thứ hai chứa n số nguyên $x_1, x_2, \dots, x_n (-1 \leq x_i \leq n + 1)$.

Gọi N là tổng giá trị của n trong các bộ dữ liệu của một file input. Dữ liệu vào đảm bảo $N \leq 4194304$.

Output

Với mỗi bộ dữ liệu, in ra kết quả như sau:

- Dòng thứ nhất chứa từ Yes nếu tồn tại một hoán vị thỏa mãn điều kiện của đề bài, in ra No nếu không tồn tại hoán vị như vậy.
- Nếu dòng thứ nhất chứa từ Yes, dòng thứ hai chứa n số nguyên p_1, p_2, \dots, p_n mô tả một hoán vị như vậy. Nếu có nhiều đáp án, bạn được phép in ra một hoán vị bất kì.

Scoring

Bộ test của bài được chia làm 5 subtask như sau:

- Subtask 1 (20 điểm): $n \leq 8$ và $N \leq 512$
- Subtask 2 (20 điểm): Với mọi $i, x_i \neq -1$.
- Subtask 3 (20 điểm): $n \leq 2048$ và $N \leq 16384$
- Subtask 4 (20 điểm): $n \leq 131072$ và $N \leq 1048576$
- Subtask 5 (20 điểm): Không có ràng buộc gì thêm.

Với mỗi test, nếu bạn đáp đúng c bộ dữ liệu đầu tiên nhưng đáp sai ở bộ dữ liệu thứ $c + 1$, bạn được $P \cdot \frac{c}{\tau}$ điểm, trong đó P là điểm tối đa của một test, được tính bằng 100 chia cho số test.

input
1 3 5 6 3 -1 5 -1 4 -1 -1 -1 -1 3 3 4 -1
output
Yes 5 1 2 3 4 Yes 1 2 3 4 No

Trong bộ dữ liệu thứ nhất, từ hoán vị $(5, 2, 3, 1, 4)$ ta có dãy $next = (6, 3, 5, 5, 6)$.

Trong bộ dữ liệu thứ hai, do $x_1 = x_2 = x_3 = x_4 = -1$ nên dãy $next$ có thể nhận giá trị bất kì. Do đó mọi hoán vị độ dài 4 đều thỏa mãn.

Trong bộ dữ liệu thứ ba, một hoán vị thỏa mãn cần có $next_1 = 3$ (tương đương với $p_2 < p_1 < p_3$) và $next_2 = 4$ (tương đương với $p_3 < p_2$). Để thấy hai ràng buộc này mâu thuẫn nhau, nên không tồn tại hoán vị nào thỏa mãn.

Statement
is not
available
on
English
language

I. Trò chơi trên bảng

0.75 seconds, 256 megabytes

Trong một lần đi chơi công viên, bé Thích Ăn Kẹo chơi một trò chơi như sau:

Bé được phát một khay đựng kẹo. Khay có dạng một bảng hình chữ nhật với r hàng và c cột. Các hàng của khay được đánh số từ 1 đến r và các cột được đánh số từ 1 đến c . Giao của mỗi hàng và mỗi cột là một ô vuông có thể đựng được tối đa một viên kẹo, ô nằm trên hàng i và cột j được kí hiệu là (i, j) .

Ban đầu trên khay của bé không có kẹo và bé phải mua một số kẹo để đặt vào các ô. Giá tiền cho viên kẹo đặt vào ô (i, j) là $p_{i,j}$. Sau đó, công viên sẽ thưởng kẹo cho bé theo nguyên tắc sau: Xét một hình chữ nhật con bất kì của khay với ít nhất hai hàng và hai cột, nếu ba trong bốn ô vuông góc của hình chữ nhật này đã có kẹo, bé sẽ được tặng thêm một viên kẹo để đặt vào ô góc còn lại. Nói cách khác, trong mọi thời điểm, nếu tồn tại bốn số nguyên x_1, x_2, y_1, y_2 sao cho $1 \leq x_1 < x_2 \leq r$ và $1 \leq y_1 < y_2 \leq c$ và chính xác ba trong bốn ô $(x_1, y_1), (x_1, y_2), (x_2, y_1)$ và (x_2, y_2) đã có kẹo, bé sẽ được tặng một viên kẹo để đặt vào ô còn lại.

Do bé Thích Ăn Kẹo rất thích ăn kẹo, bé muốn mua một số viên kẹo đặt vào bảng sao cho sau khi được thưởng, bé có đầy đủ $r \cdot c$ viên kẹo đặt vào tất cả các ô trên khay. Hãy tính tổng số tiền nhỏ nhất bé cần mua kẹo để đạt được điều này.

Input

Dòng đầu tiên chứa hai số nguyên dương r và $c (1 \leq r \cdot c \leq 10^6)$, lần lượt là số hàng và số cột của khay.

Trong r dòng còn lại, dòng thứ i chứa c số nguyên $p_{i,1}, p_{i,2}, \dots, p_{i,c} (1 \leq p_{i,j} \leq 10^6)$ thể hiện chi phí mua kẹo đặt vào các ô thuộc hàng thứ i .

Output

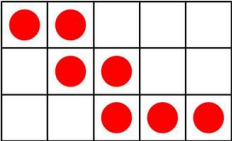
In ra một số nguyên duy nhất là số tiền nhỏ nhất cần dùng để mua kẹo sao cho sau khi được thưởng, bé Thích Ăn Kẹo có kẹo trên tất cả các ô trên khay của mình.

Scoring

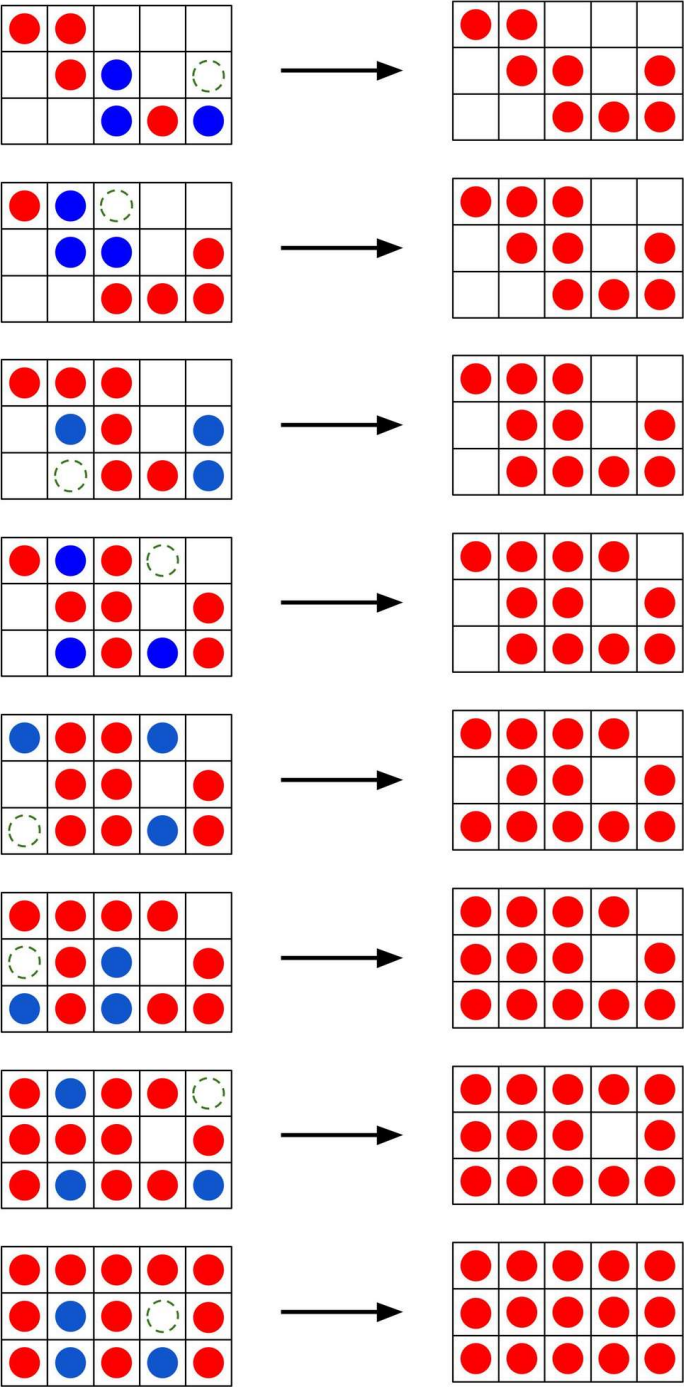
- Subtask 1 (23 điểm): $r \cdot c \leq 16$
- Subtask 2 (15 điểm): $r = 1$
- Subtask 3 (18 điểm): $r = 2$
- Subtask 4 (21 điểm): $p_{i,j} = 1$ với mọi $1 \leq i \leq r$ và $1 \leq j \leq c$.
- Subtask 5 (23 điểm): Không có ràng buộc gì thêm.

input
3 5
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
output
7

Trong ví dụ trên, đầu tiên bé Thích Ăn Kẹo mua 7 viên kẹo đặt vào các ô như sau:



Sau đó, bé lần lượt được tặng 8 viên kẹo theo thứ tự như sau:



Statement is not available on English language

J. Đường đi ngắn nhất

2 seconds, 512 megabytes

Cho một đồ thị vô hướng gồm n đỉnh và m cạnh. Các đỉnh được đánh số từ 1 đến n . Mỗi cạnh có một trọng số riêng.

Cho hai đỉnh s và t , hãy tìm độ dài đường đi ngắn nhất từ s đến t hoặc nói rằng không tồn tại đường đi giữa hai đỉnh này.

Input

Dòng đầu tiên chứa số nguyên θ ($1 \leq \theta \leq 5$) là số thứ tự của subtask chứa test này.

Dòng thứ hai chứa hai số nguyên n và m ($1 \leq n \leq 100000, 0 \leq m \leq 125000$), lần lượt là số đỉnh và số cạnh của đồ thị.

Trong m dòng tiếp theo, mỗi dòng chứa ba số nguyên u, v và c ($1 \leq u, v \leq n, 0 \leq c \leq 150000$) cho biết trên đồ thị có một cạnh nối hai đỉnh u và v với trọng số 2^c .

Dòng cuối cùng chứa hai số nguyên s và t ($1 \leq s, t \leq n$) là hai đỉnh xuất phát và kết thúc của đường đi cần tìm.

Output

Nếu không tồn tại đường đi từ s đến t , in ra -1 . Ngược lại, in ra một số nguyên duy nhất là độ dài đường đi ngắn nhất từ s đến t . Do kết quả có thể rất lớn, bạn chỉ cần in ra phần dư của độ dài đường đi ngắn nhất khi chia cho 998244353.

Scoring

- Subtask 1 (11 điểm): $c \leq 40$
- Subtask 2 (15 điểm): $c \leq 200$
- Subtask 3 (23 điểm): $n \leq 3000$ và $c \leq 3000$
- Subtasks 4 (24 điểm): Các giá trị c đôi một phân biệt.
- Subtask 5 (27 điểm): Không có ràng buộc gì thêm.

input
1 4 4 1 2 0 2 3 0 3 4 0 1 4 2 1 4
output
3

Statement
is not
available
on
English
language

NAD1. Chia đồ thị

1 second, 256 megabytes

Nếu bạn gặp khó trong việc xem đề bài, hãy tải file `split-materials.zip` trong mục `Contest materials` và xem các file ảnh bên trong để có bản đề đầy đủ và đẹp nhất.

Thí sinh chỉ có thể giải bài toán này bằng ngôn ngữ C/C++.

Alob và Bice là đôi bạn thân đến từ bộ tộc C. Trong bộ tộc này, mỗi người sẽ hoặc có mắt màu đỏ hoặc có mắt màu xanh. Người có mắt màu xanh thì luôn thích các số chẵn, còn người có mắt màu đỏ thì luôn thích các số lẻ. Biết rằng **Alob có mắt màu xanh**.

Problems - Codeforces

Một ngày, hai bạn nhận được một đồ thị vô hướng gồm n đỉnh, các đỉnh được đánh số từ 0 đến $n - 1$. Hai bạn muốn chia đồ thị này ra làm 2 phần nhỏ, mỗi phần thuộc về một bạn, theo cách như sau:

- Mỗi đỉnh trong số n đỉnh của đồ thị ban đầu thuộc về **chính xác** một trong hai bạn.
- Một trong hai bạn có thể không nhận được đỉnh nào.
- Mỗi bạn sẽ nhận về các cạnh trong đồ thị gốc thoả mãn tính chất: hai đầu của cạnh đều là hai đỉnh thuộc về bạn đó. Các cạnh trong đồ thị gốc có hai đầu mút là hai đỉnh thuộc về hai bạn khác nhau sẽ bị loại bỏ.

Vì sở thích số chẵn và số lẻ của bộ tộc này, hai bạn đều muốn nhận về các phần đồ thị thoả mãn các tính chất dưới đây:

- Bạn có mắt màu xanh muốn trong phần đồ thị mình nhận về **tất cả các đỉnh** đều có bậc chẵn.
- Bạn có mắt màu đỏ muốn trong phần đồ thị mình nhận về **tất cả các đỉnh** đều có bậc lẻ.

Hãy tìm ra một cách chia đồ thị thoả mãn các yêu cầu trên, hoặc thông báo rằng việc chia đồ thị này là không thể.

Lưu ý rằng, đồ thị được cho có thể chứa cạnh lặp và khuyên. Khuyên là cạnh nối một đỉnh tới chính nó, và trong trường hợp này, bậc của đỉnh có khuyên **được tăng thêm 2**.

CHI TIẾT CÀI ĐẶT

Thí sinh cần cài đặt hàm `vector<int> split(const string &color, int n, const vector<int> &x, const vector<int> &y)`. Các tham số của hàm có ý nghĩa như sau:

- `color` là màu mắt của Bice.
- `n` là số đỉnh của đồ thị.
- Hai vector `x` và `y` mô tả danh sách cạnh của đồ thị. Cả hai vector cùng có số phần tử là m . Với mỗi chỉ số i từ 0 đến $m - 1$, đồ thị có một cạnh nối hai đỉnh `x[i]` và `y[i]`.

Hàm `split` trả về một vector chứa chỉ số của các đỉnh trong phần của Alob. Trong trường hợp không có cách chia đồ thị thoả mãn, hàm `split` trả về vector `{=1}`. Nếu có nhiều cách chia thoả mãn, bạn được trả về một cách bất kỳ.

Lưu ý:

- Thí sinh cần có dòng lệnh khai báo thư viện `#include "split.h"` ở dòng đầu tiên của chương trình.
- File mã nguồn của thí sinh không được chứa hàm `main`. Việc viết hàm `main` có thể gây ra lỗi biên dịch cho bài làm của thí sinh.
- Thí sinh không được đọc dữ liệu từ thiết bị vào chuẩn hay bất cứ tệp tin nào, cũng như không được in ra thiết bị ra chuẩn hay bất kì tệp tin nào.
- Chương trình của thí sinh được phép khai báo thêm các biến toàn cục nếu cần.
- Vui lòng tham khảo chương trình mẫu (file `split-sample.cpp` trong file nén `split-materials.zip` có ở mục `Contest materials`) để biết thêm chi tiết.

Scoring

Trong tất cả các test, các tham số truyền vào hàm `split` thoả mãn:

- `color` chỉ nhận một trong hai giá trị `blue` (mắt màu xanh) hoặc `red` (mắt màu đỏ).
- `n` có giá trị từ 1 đến 500.
- Hai vector `x` và `y` có từ 0 đến 1500 phần tử.

- Các phần tử của hai vector x và y có giá trị từ 0 đến $n - 1$.
Dưới đây là một số ví dụ:

- Hàm `split("blue", 7, {}, {})`: đồ thị có 7 đỉnh nhưng không có cạnh nào. Vì vậy mọi đỉnh đều có bậc chẵn (bằng 0). Cả Alob và Bice đều có mất màu xanh, nên đều muốn các đỉnh trong đồ thị của mình có bậc chẵn. Do đó, mọi cách chia đều hợp lệ. Một trong các giá trị trả về chính xác là $\{2, 3, 5\}$.
- Hàm `split("red", 8, {6, 1, 4, 7, 7, 5, 2, 3, 0, 6, 1}, {3, 0, 2, 4, 4, 4, 6, 5, 5, 4, 1})` trả về $\{1, 3, 4, 7\}$. Trong trường hợp này, Alob có các đỉnh 1, 3, 4, 7 cùng các cạnh (1, 1), (4, 7) và (4, 7). Như vậy, bậc của các đỉnh đều chẵn: $deg_1 = deg_4 = deg_7 = 2$ và $deg_3 = 0$. Bice có các đỉnh 0, 2, 5, 6 cùng các cạnh (0, 5) và (2, 6). Như vậy, cả 4 đỉnh mà Bice nhận được cùng có bậc là 1 (là số lẻ). Điều này thoả mãn yêu cầu của Bice do Bice có mất màu đỏ.

Statement
is not
available
on
English
language

NAD2. Đếm lát gạch

6 seconds, 256 megabytes

Cho một bảng hình chữ nhật gồm r hàng và c cột. Bạn cần đục lỗ một số ô trên bảng, sao cho với những ô còn lại, tồn tại ít nhất một cách lát gạch các ô này bằng các viên gạch kích thước 1×2 và 2×1 . Một cách lát gạch được coi là hợp lệ nếu mỗi ô không bị đục lỗ được phủ bởi **chính xác** một viên gạch.

Do kết quả có thể rất lớn, bạn cần in ra theo modulo $10^9 + 7$.

Input

Problems - Codeforces

Gồm một dòng duy nhất chứa hai số nguyên r và c ($1 \leq r \leq 8, 1 \leq c \leq 8000$). Dữ liệu vào đảm bảo:

- Nếu $r = 8$ thì $c \leq 200$
- Nếu $r = 7$ thì $c \leq 1000$

Output

Gồm một số nguyên duy nhất là phần dư của số cách đục lỗ hợp lệ khi chia cho $10^9 + 7$.

Scoring

- Subtask 1 (15 điểm): $r \cdot c \leq 20$
- Subtask 2 (10 điểm): $r = 1$
- Subtask 3 (15 điểm): $r = 2$
- Subtask 4 (15 điểm): $r \leq 4$
- Subtask 5 (15 điểm): $r \leq 6$
- Subtask 6 (15 điểm): $r = 7$
- Subtask 7 (15 điểm): $r = 8$

input
2 3
output
18

Dưới đây là những cách đục lỗ hợp lệ trong trường hợp $r = 2$ và $c = 3$. Ký tự x thể hiện những ô bị đục lỗ:

```
xxx  xx.  x.x  .xx  ..x  x..  xxx  xxx  ...  
xxx  xx.  x.x  .xx  xxx  xxx  ..x  x..  ...  
  
..x  .x.  x..  x..  ..x  xx.  .xx  ...  ...  
..x  .x.  x..  ..x  x..  ...  ...  xx.  .xx
```