

101. Đuổi bò

Program RUNAWAY.*
 Input RUNAWAY.INP
 Output RUNAWAY.OUT
 Score 100

Đã tới giờ cho uống nước ở nông trại của nông dân John (FJ), nhưng các con bò lại đang bỏ chạy! FJ muốn tập trung chúng lại, và ông ta cần sự giúp đỡ của bạn. Nông trại của FJ là một dãy gồm có N ($1 \leq N \leq 200000$) bãi cỏ được đánh số từ $1..N$ và được nối bằng $N-1$ con đường hai chiều. Chuồng bò nằm ở bãi cỏ thứ nhất, và từ bãi thứ nhất, ta có thể đi đến tất cả các bãi cỏ còn lại. Những con bò của FJ đang ở bãi cỏ của chúng vào sáng nay, nhưng không ai biết chúng đã đi đâu cho tới bây giờ. FJ biết rằng những con bò chỉ muốn chạy xa khỏi nhà chứa, nhưng cũng rất lười nên không thể chạy một đoạn đường có độ dài lớn hơn L (theo hướng xa nhà chuồng). Với mỗi bãi cỏ, FJ muốn biết có bao nhiêu bãi cỏ mà những con bò bắt đầu tại bãi cỏ đó có thể dừng chân.

Lưu ý: Số dạng 64 bit (trong Pascal là int64, trong C/C++ là long long, và trong Java là long) cần dùng để lưu các khoảng cách.

Input:

- Dòng đầu tiên ghi hai số nguyên N, L ($1 \leq N \leq 200000, 1 \leq L \leq 10^{18}$)
- $N-1$ dòng tiếp theo, dòng thứ i ghi hai số p_i, l_i với p_i là bãi cỏ đầu tiên trên đường đi ngắn nhất từ $i + 1$ đến nhà chứa, l_i là khoảng cách con đường đó ($1 \leq p_i < i + 1, 1 \leq l_i \leq 10^{12}$)

Output: gồm N dòng, dòng thứ i là số lượng bãi cỏ có thể đi tới được từ i bằng cách rời xa nhà chứa (bãi cỏ 1) với độ dài không quá L .

Example:

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

Giải thích:

- Con bò ở bãi cỏ 1 (chuồng) có thể ở bãi 1, 2, 4
- Con bò ở bãi cỏ 2 có thể ở bãi 2, 3
- Các con bò ở bãi 3 và 4 không thể đi xa hơn nữa khỏi bãi 1 (chuồng) nên nó chỉ có thể ở đó

102. Xếp lịch hoàn thành công việc

Program ORDER.*
 Input ORDER.INP
 Output ORDER.OUT
 Score 100

Có N công việc được đánh số từ 1 đến N . Hoàn thành công việc i phải chi phí cho thợ số tiền là p_i (gọi là giá gốc của công việc i). Ngoài ra nếu công việc j được hoàn thành trước công việc i thì muốn thợ hoàn thành công việc i phải trả thêm cho thợ là s_{ij} đồng nữa (để đảm bảo khi tiến hành công việc i , thợ không phá vỡ hoặc làm hỏng công việc j). Ví dụ, khi trang trí nội thất, nếu trong nhà chưa lắp cửa kính thì chỉ phải trả tiền sơn tường là 10 triệu đồng, nhưng khi đã lắp cửa kính rồi mới sơn nhà thì phải trả thêm cho thợ sơn nhà một khoản phí là 2 triệu đồng nữa).

Hãy viết chương trình xếp lịch làm các công việc sao cho chi phí tiền công thợ là nhỏ nhất.

Input:

- Dòng đầu tiên chứa số nguyên dương N ($0 < N < 16$).
- N dòng tiếp theo, mỗi dòng chứa đúng N số nguyên: số nguyên thứ i là giá gốc của công việc thứ i , số nguyên thứ j trên dòng này (i khác j) là số tiền s_{ij} phải trả thêm cho công việc i nếu công việc j đã hoàn thành trước công việc i . Các giá tiền nguyên không âm, không vượt quá 100000.

Output: Một số nguyên duy nhất là tổng số tiền nhỏ nhất để hoàn thành N công việc.

Example:

Input	Output
2 10 10 9000 10	30
3 14 23 0 5 14 0 1000 9500 14	47

103. Thường thức

Program ENJOY.*
Input ENJOY.INP
Output ENJOY.OUT
Score 100

Khi BT bước vào nhà hàng và ngồi xuống bàn, nhân viên phục vụ ngay lập tức mang cho anh ta thực đơn. Có n món ăn và BT muốn ăn đúng m món trong số chúng.

BT biết rằng món thứ i sẽ đem lại cho anh ta độ dễ chịu là a_i . Có một số món ăn không được ăn cùng nhau và một số món sẽ trở lên tuyệt vời nếu ăn cùng nhau. BT có k quy tắc tăng độ dễ chịu có dạng như sau: Nếu ăn món ăn x chính xác trước món ăn y (tức là không ăn món ăn nào khác giữa x và y) thì độ dễ chịu tăng lên một lượng là c .

Hãy giúp BT chọn các món ăn và trình tự ăn sao cho tổng độ dễ chịu thu hoạch được là lớn nhất

Input:

- Dòng đầu tiên chứa ba số nguyên n, m, k ($1 \leq m \leq n \leq 18, 0 \leq k \leq n \cdot (n - 1)$) - số lượng món ăn trong thực đơn, số món ăn cần ăn và số quy tắc tăng độ dễ chịu.
- Dòng thứ hai chứa n số nguyên a_i ($0 \leq a_i \leq 10^9$) - độ dễ chịu của món ăn thứ i
- k dòng tiếp theo mỗi dòng chứa một qui tắc. Dòng thứ i ghi ba số nguyên x_i, y_i và c_i ($1 \leq x_i, y_i \leq n; 0 \leq c_i \leq 10^9$) có nghĩa rằng nếu ăn món ăn x_i ngay trước món ăn y_i thì độ dễ chịu tăng lên một lượng là c_i . Chú ý rằng không tồn tại hai cặp chỉ số i, j thỏa mãn $1 \leq i < j \leq k$ mà $x_i = x_j$ và $y_i = y_j$.

Output: Một số nguyên duy nhất là tổng độ dễ chịu lớn nhất có được.

Example:

Input	Output
4 3 2 1 2 3 4 2 1 5 3 4 2	12

104. Hôn nhân

Program MARRIAGE.*
Input MARRIAGE.INP
Output MARRIAGE.OUT
Score 100

Ở một làng nọ có n cô gái chưa chồng xinh đẹp. Vào một ngày đẹp trời có n chàng trai tuấn tú đến chỗ già bản xin lấy n cô gái làm vợ (mỗi chàng trai lấy một cô gái). Già bản vốn là một người thông thái nhìn xa trông rộng nên dự tính được nếu chàng trai thứ i lấy cô gái thứ j làm vợ thì làng sẽ nhận được số tiền hồi môn là a_{ij} .

Hỏi rằng già bản bố trí n đám cưới như thế nào (mỗi chàng trai lấy một cô gái và mỗi cô gái chỉ lấy một chàng trai) sao cho tổng số tiền hồi môn mà dân làng nhận được là lớn nhất. Bạn cũng chỉ cần tính số tiền này. Ngoài ra cũng cần phải biết có bao nhiêu cách xếp khác nhau để được số tiền trên

(hai cách xếp là khác nhau nếu như có ít nhất một cặp xuất hiện trong cách này nhưng không xuất hiện trong cách kia)

Input:

- Dòng đầu tiên ghi số nguyên dương $n \leq 20$ là số cô gái và số chàng trai.
- n dòng tiếp theo, dòng thứ i ghi n số, số thứ j là a_{ij} ($0 \leq a_{ij} \leq 10^7$)

Output: In ra hai số nguyên là tổng số tiền hồi môn lớn nhất và số cách để có được số tiền như vậy

Example:

Input	Output
4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	34 24

105. Robot lau nhà

Program MCLEAN.*
 Input MCLEAN.INP
 Output MCLEAN.OUT
 Score 100

Sàn nhà là một hình chữ nhật được chia thành các ô. Các ô của sàn nhà có thể là ô sạch, ô bẩn hoặc có vật cản. Robot có thể dọn một ô bẩn thành ô sạch nếu như nó ở trên ô đó. Mỗi bước, robot có thể từ một ô di chuyển đến một trong 4 ô chung cạnh nếu như ô này không có vật cản.

Xác định số bước ít nhất để một robot có thể lau sạch các ô bẩn trên sàn nhà (nếu có thể).

Input:

- Dòng đầu tiên ghi n, m là số cột và số hàng của sàn nhà ($1 \leq n, m \leq 20$)
- m dòng tiếp theo, mỗi dòng ghi n ký tự mô tả tình trạng của các ô vuông trên sàn nhà với:
 - '.' : ô sạch
 - '*' : ô bẩn
 - 'x' : ô có vật cản
 - 'o' : ô xuất phát của robot (1 con)

Có không quá 10 ô bẩn

Output: In ra số bước di chuyển ít nhất để robot có thể lau sạch các ô bẩn. In ra -1 nếu như nó không thể làm được điều này

Example:

Input	Output
7 5o...*.*...*.	8

106. T-Shirts

Program TSHIRTS.*
 Input TSHIRTS.INP
 Output TSHIRTS.OUT
 Score 100

Cinderella cùng với Hoàng tử tổ chức một bữa tiệc nhỏ. Mỗi người đến dự tiệc đều mang theo một số chiếc áo thun. Có tất cả 100 loại áo thun khác nhau và được đánh số từ 1 đến 100. Tất nhiên không có người nào có hai chiếc áo thun cùng loại.

Cinderella muốn biết có bao nhiêu cách khác nhau để những người dự tiệc mặc chiếc áo thun của mình sao cho không có hai người mặc cùng một loại áo thun. Hai cách mặc được gọi là khác nhau nếu như có ít nhất một người mặc hai loại áo thun khác nhau.

Input: Dòng đầu tiên ghi số nguyên dương $T \leq 10$ là số bộ dữ liệu. Tiếp theo là T nhóm dòng mô tả các bộ dữ liệu với cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên dương $n \leq 10$ là số người có mặt trong bữa tiệc.
- n dòng tiếp theo, dòng thứ i chứa ít nhất một số nguyên và nhiều nhất 100 số nguyên khác nhau mô tả các loại áo thun mà người thứ i mang theo.

Output: Với tất cả các bộ dữ liệu in ra số lượng cách mặc áo. Con số này có thể rất lớn nên bạn chỉ cần lấy phần dư của nó khi chia cho 10^9+7

Example:

input	output
2	4
2	4
3 5	
8 100	
3	
5 100 1	
2	
5 100	

107. Cắt bánh

Program PARTITION.*
 Input PARTITION.INP
 Output PARTITION.OUT
 Score 100

Chiếc bánh sinh nhật của Mai có dạng hình vuông được chia thành $N \times N$ ô vuông. Trên mỗi ô vuông có gắn một số lượng sô-cô-la khác nhau.

Theo truyền thống Mai thực hiện cắt bánh không quá K nhát cắt song song với các cạnh của bánh và chỉ đi qua cạnh các ô vuông ($1 \leq K \leq 2N - 2$)

Mai muốn rằng sau khi cắt bánh thành các phần thì lượng sô-cô-la ở phần có nhiều sô-cô-la nhất là ít nhất. Viết chương trình giúp cô bé thực hiện điều này.

Input:

- Dòng đầu tiên ghi hai số nguyên dương N, K ($2 \leq N \leq 15$)
- N dòng tiếp theo, dòng thứ i ghi N số nguyên nằm trong khoảng từ 0 đến 1000 mô tả lượng sô-cô-la ở các ô tương ứng trên hàng thứ i (tính từ trái qua phải)

Output: Một số nguyên là lượng sô-cô-la nhỏ nhất ở phần có nhiều sô-cô-la nhất.

Example:

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

108. Remembering Strings

Program REMEMBER.*
 Input REMEMBER.INP
 Output REMEMBER.OUT
 Score 100

Bạn có n xâu ký tự có cùng độ dài chỉ chứa các chữ cái tiếng Anh thường. Chúng ta nói rằng các xâu ký tự này là dễ nhớ nếu như với mỗi xâu ký tự đều tồn tại một vị trí i và chữ cái c sao cho chỉ dãy ký tự này có chữ cái c ở vị trí i .

Ví dụ các dãy ký tự {"abc", "aba", "adc", "ada"} không phải là dễ nhớ. Tuy nhiên tập các dãy {"abc", "ada", "ssa"} là dễ nhớ bởi vì:

- Dãy đầu tiên là dãy duy nhất ở vị trí 3 có ký tự 'c'
- Dãy thứ hai là dãy duy nhất ở vị trí 2 có ký tự 'd'
- Dãy thứ ba là dãy duy nhất ở vị trí 1 có ký tự 's'

Bạn có thể thay đổi tập các dãy ký tự một chút cho dễ nhớ. Với số tiền là a_{ij} bạn có thể đổi ký tự ở vị trí thứ j của xâu ký tự thứ i thành một ký tự in thường tiếng Anh bất kỳ.

Hãy tính số tiền tối thiểu bạn cần phải có để có thể biến đổi tập các dãy ký tự đã cho sao cho dễ nhớ

Input:

- Dòng đầu tiên chứa hai số nguyên dương n, m ($1 \leq n, m \leq 20$) lần lượt là số lượng dãy ký tự và độ dài của mỗi dãy ký tự.
- n dòng tiếp theo, mỗi dòng chứa một dãy ký tự của tập. Các dãy này có cùng độ dài m
- n dòng cuối cùng, dòng thứ i chứa m số nguyên $a_{i1}, a_{i2}, \dots, a_{im}$ ($0 \leq a_{ij} \leq 10^6$)

Output: In ra một số nguyên duy nhất là số tiền ít nhất phải có

Example:

Input	Output
4 5 abcde abcde abcde abcde 1	3

109. Ném đĩa

Program MGUARD.*
Input MGUARD.INP
Output MGUARD.OUT
Score 100

Nông dân John(FJ) và đàn bò của ông ta đang chơi môn ném đĩa. Bessie ném cái đĩa xuống sân, nhưng nó lại đi thẳng tới vị trí của Mark ở bên đối thủ. Mark có chiều cao H ($1 \leq H \leq 10^9$), nhưng có N con bò ở bên đội của Bessie đang đứng quanh Mark ($2 \leq N \leq 20$). Những con bò này chỉ có thể bắt được cái đĩa nếu chúng có thể đứng chồng lên nhau với độ cao ít nhất phải bằng độ cao của Mark. Mỗi con bò N có một độ cao, cân nặng, và sức khỏe khác nhau. Sức khỏe của một con bò được xác định bằng số lượng cân nặng tối đa mà nó có thể chất những con bò khác lên trên nó. Được cho trước những thông tin trên, Bessie muốn biết rằng đội của cô ta có thể bắt được cái đĩa hay không, và nếu bắt được thì Bessie muốn biết hệ số an toàn tối đa của các con bò chất lên nhau đó. Hệ số an toàn tối đa của một đồng con bò chồng lên nhau là số lượng cân nặng có thể chồng lên trên các con bò này, với điều kiện rằng không có con bò nào phải giữ các con bò ở trên vượt quá sức chịu đựng của mình.

Input:

- Dòng thứ nhất gồm hai số N và H .
- N dòng tiếp theo, mỗi dòng mô tả một con bò: chiều cao, cân nặng, và sức khỏe. Tất cả các số này đều là số nguyên dương và không vượt quá 1 tỉ.

Output: Nếu đội của Bessie có thể chất thành một đồng đủ cao để bắt được cái đĩa, hãy in ra hệ số an toàn tối đa có thể đạt được. Nếu không, hãy in ra "Mark is too tall" (không bao gồm dấu mở/đóng ngoặc kép).

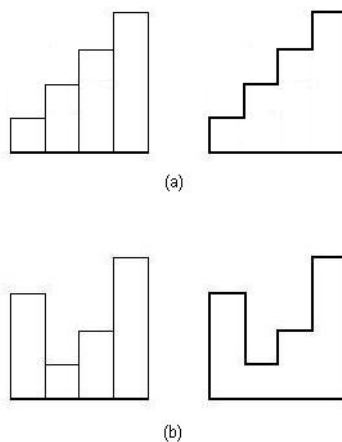
Example:

input	output
4 10 9 4 1 3 3 5 5 5 10 4 4 5	2

110. Biểu đồ

Program	HISTOGRAM.*
Input	HISTOGRAM.INP
Output	HISTOGRAM.OUT
Score	100

Trong thống kê người ta hay sử dụng biểu đồ cột đứng gồm những hình chữ nhật độ rộng đơn vị đứng liền nhau. Ta có một biểu đồ như vậy với chiều cao của các hình chữ nhật khác nhau từng đôi một. Một vài biểu đồ sẽ cho chu vi bao quanh các hình chữ nhật lớn nhất. Nhiệm vụ của bạn là tìm hoán vị các cột để cho chu vi của biểu đồ là lớn nhất và đếm xem có bao nhiêu hoán vị như vậy?



Trong hình trên, hình (a) ứng với hoán vị (1,2,3,4) (bộ dữ liệu 1 trong ví dụ) có chu vi là 16 còn hình (b) ứng với hoán vị (3,1,2,4) có chu vi là 20 (đây cũng là giá trị lớn nhất).

Input: Gồm nhiều bộ dữ liệu, mỗi bộ dữ liệu bắt đầu bằng một dòng chứa số nguyên dương n ($2 \leq n \leq 15$). Dòng thứ hai gồm n số nguyên dương phân biệt mô tả độ cao các cột (độ cao có giá trị không quá 100). Giá trị $n = 0$ báo hiệu kết thúc các bộ dữ liệu. Trong mỗi file input có không quá 50 bộ dữ liệu.

Output: Ứng với mỗi bộ dữ liệu in ra một dòng chứa hai số nguyên, số thứ nhất là chu vi lớn nhất và số thứ hai là số lượng hoán vị cho kết quả là chu vi này.

Example:

input	output
4 1 2 3 4 3 2 6 5 0	20 8 24 2

Subtasks:

- Subtask 1: $n \leq 10$
- Subtask 2: $n \leq 15$

111. Đếm số lượng tổ hợp

Program	CKN.*
Input	CKN.INP
Output	CKN.OUT

Score 100

Giả sử các tổ hợp chập k của $\{1, 2, \dots, n\}$ được viết dưới dạng (x_1, x_2, \dots, x_k) với $1 \leq x_1 < x_2 < \dots < x_k \leq n$ và được sắp xếp theo thứ tự từ điển.

1. Đếm số lượng các tổ hợp
2. Cho một tổ hợp, tìm số hiệu của nó trong từ điển
3. Cho một số hiệu tìm tổ hợp tương ứng

Input:

- Dòng đầu ghi số N K ($1 \leq K \leq N \leq 100$)
- Tiếp theo là một số dòng, mỗi dòng có một trong hai dạng sau:
 - 1 $x_1 x_2 \dots x_k$ thể hiện một tổ hợp, yêu cầu tìm số thứ tự tương ứng
 - 2 P thể hiện một số thứ tự, yêu cầu tìm tổ hợp tương ứng

Output:

- Dòng đầu ghi số lượng tổ hợp tìm được
- Các dòng tiếp theo tương ứng với các câu trả lời đối với một dòng trong file dữ liệu vào. Nếu là loại 1 ... thì cho ra số thứ tự tương ứng, nếu là loại 2 ... thì cho ra tổ hợp tương ứng.

Example:

input	output
7 4	35
1 2 4 6 7	29
2 1	1 2 3 4

Ghi chú: Có 60% số test có $N \leq 60$

112. Số lượng tập con

Program TAPCON.*
 Input TAPCON.INP
 Output TAPCON.OUT
 Score 100

Cho tập hợp $\{1, 2, \dots, n\}$. Người ta liệt kê các tập con (khác rỗng) của tập này theo thứ tự từ điển (với mỗi tập con khác rỗng, các phần tử của nó được liệt kê tăng dần). Ví dụ, với $n=3$ ta có 7 tập con đánh số như sau:

1: 1	5: 2
2: 1 2	6: 2 3
3: 1 2 3	7: 3
4: 1 3	

1. Đếm số lượng các tập hợp con
2. Cho một tập con, tìm số thứ tự của nó
3. Cho một số thứ tự, tìm tập con tương ứng.

Input:

- Dòng đầu ghi số N ($N \leq 100$)
- Tiếp theo là một số dòng, mỗi dòng có một trong hai dạng sau:
 - 1 $x_1 x_2 \dots x_k$ thể hiện một tập con, tìm số thứ tự tương ứng
 - 2 P thể hiện một số thứ tự, yêu cầu tìm tập con tương ứng

Output:

- Dòng đầu ghi số lượng tập con tìm được
- Các dòng tiếp theo tương ứng với các câu trả lời đối với một dòng trong file dữ liệu vào. Nếu là loại 1 ... thì cho ra số thứ tự tương ứng, nếu là loại 2 ... thì cho ra tập con tương ứng.

Example:

input	output
3	7
1 1 2 3	3
6	2 3

113. Số catalan

Program	CATALAN.*
Input	CATALAN.INP
Output	CATALAN.OUT
Score	100

Với N cho trước, xét các dãy $A=(a_1, a_2, \dots, a_{2n})$ trong đó:

- a_i nguyên không âm
- $a_0=a_{2n}=0$
- $|a_i - a_{i+1}| = 1$

Số lượng các dãy số A thỏa mãn tính chất trên là số Catalan bậc N. Các dãy số A có thể được sắp theo thứ tự từ điển. Theo thứ tự từ điển này, mỗi số có một số hiệu.

Ví dụ, với $N=3$ ta có 5 dãy số

- 1) 0 1 0 1 0 1 0
- 2) 0 1 0 1 2 1 0
- 3) 0 1 2 1 0 1 0
- 4) 0 1 2 1 2 1 0
- 5) 0 1 2 3 2 1 0

Dãy 1 2 1 2 1 0 có số hiệu 4

Yêu cầu: Tìm số Catalan theo N, $N \leq 100$. Biết dãy số A, tìm số hiệu của nó trong từ điển và ngược lại

Input:

- Dòng đầu tiên ghi số nguyên dương N
- Các dòng sau có một trong 2 dạng
 - 1 $a_0 \dots a_{2n}$: Tìm số hiệu ứng với dãy A đã biết
 - 2 P: Tìm dãy A khi biết số hiệu

Output:

- Dòng đầu tiên ghi số Catalan bậc N
- Tiếp theo là các dòng tương ứng với các dòng của file dữ liệu vào. Nếu trong file dữ liệu vào có dạng 1... thì in ra một số thứ tự. Nếu trong file dữ liệu vào có dạng 2... thì in ra một dãy A

Example:

input	output
3	5
2 2	0 1 0 1 2 1 0
1 0 1 2 3 2 1 0	5
2 4	0 1 2 1 2 1 0

114. Đếm hoán vị

Program	HV.*
Input	HV.INP
Output	HV.OUT
Score	100

Các dãy hoán vị của $1, 2, \dots, n$ được sắp xếp theo thứ tự từ điển.

1. Đếm số lượng các hoán vị
2. Cho một hoán vị, tìm số hiệu của nó trong từ điển
3. Cho một số hiệu tìm hoán vị tương ứng

Input:

- Dòng đầu ghi số N ($N \leq 50$)
- Tiếp theo là một số dòng, mỗi dòng có một trong hai dạng sau:
 - 1 $x_1 x_2 \dots x_n$ thể hiện một hoán vị, yêu cầu tìm số thứ tự tương ứng
 - 2 P thể hiện một số thứ tự, yêu cầu tìm hoán vị tương ứng

Output:

- Dòng đầu ghi số lượng hoán vị tìm được
- Các dòng tiếp theo tương ứng với các câu trả lời đối với một dòng trong file dữ liệu vào.
Nếu là loại 1 ... thì cho ra số thứ tự tương ứng, nếu là loại 2 ... thì cho ra hoán vị tương ứng.

Example:

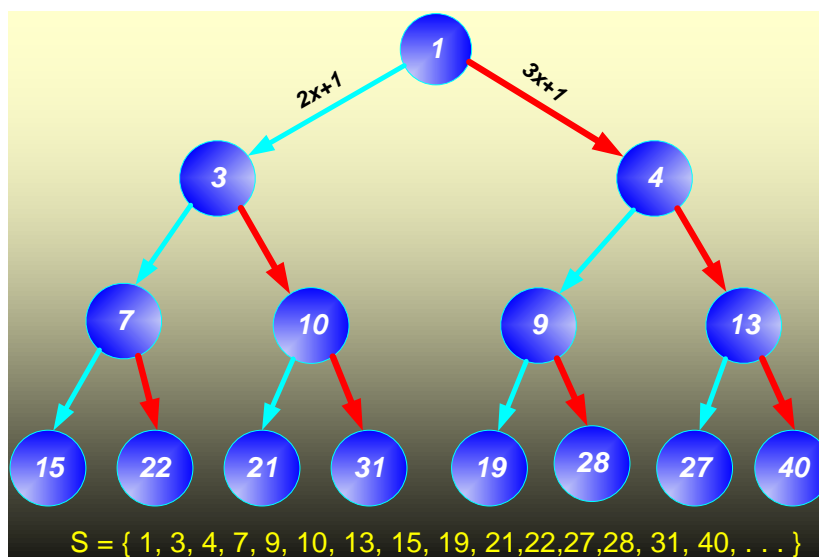
Input	Output
3	6
1 2 3 1	4
2 6	3 2 1

115. Tập số

Program SET.*
 Input SET.INP
 Output SET.OUT
 Score 100

Xét tập số nguyên S được định nghĩa như sau:

- 1 thuộc S,
- Nếu x thuộc S, thì $2x+1$ và $3x+1$ cũng thuộc S,
- S không chứa số nào khác ngoài các số theo quy tắc đã nêu.



Yêu cầu: Cho số nguyên N ($1 \leq N \leq 2^{31}-1$). Hãy xác định số thứ N của S, nếu các số trong S được sắp xếp theo thứ tự tăng dần.

Input: Gồm nhiều dòng, mỗi dòng chứa một số nguyên N.

Output: Mỗi kết quả là một số nguyên và đưa ra trên một dòng.

Example:

Input	Output
100	418
254	1461

116. Đoạn một

Program SEGONE.*
 Input SEGONE.INP
 Output SEGONE.OUT
 Score 100

Cho hai số nguyên dương N và k , tập S gồm các dãy nhị phân độ dài N thỏa mãn số các đoạn 1 liên tiếp bằng k . Ví dụ $N=5, k=2$ tập S gồm các dãy 00101, 01011, 11011,.... Các phần tử của được sắp xếp theo thứ tự tăng dần của số nguyên tương ứng. Hãy cho biết dãy nhị phân thứ M của S .

Input: Gồm một dòng 3 số nguyên N, k, M ($N \leq 50$). Các số cho đảm bảo có nghiệm.

Output: Một dòng gồm các số 0, 1 ghi liền nhau thể hiện dãy nhị phân tìm được.

Example:

input	output
5 2 2	01011

117. Cân đĩa

Program BALANCE.*
 Input BALANCE.INP
 Output BALANCE.OUT
 Score 100

Cho một cái cân đĩa và n quả cân đánh số từ 1 đến n , quả thứ i có khối lượng i . Với một vật có khối lượng m , người ta đặt vật đó vào đĩa cân bên trái và sau đó thêm một số quả cân vào hai đĩa cân sao cho cân thăng bằng. Từ đó xác định khối lượng của vật.

Hai cách cân được gọi là khác nhau nếu như tập các quả cân ở đĩa trái trong hai cách khác nhau là khác nhau hoặc tập các quả cân ở đĩa phải trong hai cách là khác nhau.

Yêu cầu: Đếm số cách cân vật đã cho.

Input: Gồm một dòng chứa hai số nguyên dương $n, m \leq 100$

Output: Một số nguyên duy nhất là số cách cân vật

Example:

Input	Output
4 2	7

118. Tung đồng xu

Program TOSS.*
 Input TOSS.INP
 Output TOSS.OUT
 Score 100

Ngày xưa, cách đây đã lâu lắm rồi, ở vương quốc Byteland tươi đẹp có một nàng công chúa xinh đẹp tuyệt trần. Thật không may, chính vì sự xinh đẹp đó đã làm phù thủy Astral đã bắt làm về làm người hầu cho ông ta. Đức Vua vô cùng hoang mang khi chuyện này xảy ra, ông không biết phải làm cách nào để giải cứu con mình (ông không thể mang quân đến đánh vì điều đó là vô nghĩa). Tuy nhiên, tên phù thủy này lại rất sợ một câu thần chú được suy ra từ việc giải một bài toán cổ của Thần Sphinx. Bài toán đó có thể được mô tả một cách đơn giản như sau: “*Khi ta tung một đồng xu, ta sẽ nhận được mặt sấp hoặc ngửa. Nếu ta tung lần lượt N đồng xu thì có bao nhiêu trường hợp mà có ít nhất K đồng xu liên tiếp cùng là ngửa?*”. Đức vua hứa sẽ thưởng rất hậu hĩnh và gả công chúa cho ai giải được bài toán này. Thực ra công chúa và anh chàng làm vườn trong hoàng cung đã yêu thương nhau từ lâu. Anh chàng giờ đây đang rất bối rối và cần sự giúp đỡ của bạn.

Input: Một dòng duy nhất ghi hai số N và K . $1 \leq K \leq N \leq 10000$

Output: Một dòng duy nhất ghi số trường hợp đếm được. Vì kết quả có thể rất lớn nên bạn chỉ cần in ra phần dư của nó khi chia cho $10^9 + 7$

Example:

input	output
4 2	8
4 3	3

119. Stripe

Program	STRIPE.*
Input	STRIPE.INP
Output	STRIPE.OUT
Score	100

Cho hình chữ nhật kích thước $1 \times N$, trong đó các hình vuông con kích thước 1×1 có thể được tô bằng màu đen hoặc tô bằng màu trắng. Bạn có thể mã hoá hình chữ nhật bằng một dãy các số, số các hình vuông màu đen liên tiếp nhau từ trái sang phải.



Ví dụ như hình trên thì dãy mã hoá là 2 3 2 8 1. Song nếu chỉ mã hoá như vậy thì sẽ có rất nhiều hình chữ nhật thoả mãn dãy số trên ví dụ như hình sau:



Hãy lập trình tính số hình chữ nhật thoả mãn dãy mã hoá cho trước.

Input:

- Dòng đầu gồm 2 số là chiều dài N của hình chữ nhật ($N \leq 200$) và K là số phần tử của dãy mã hoá ($0 \leq K \leq (N+1)/2$)
- Dòng thứ ba ghi K số nguyên mô tả dãy số mã hoá

Output: chỉ gồm một dòng duy nhất chứa một số là số hình chữ nhật thoả mãn.

Example:

input	output
5 2 1 2	3

120. Khảo sát trò chơi

Program	SEARCH.*
Input	SEARCH.INP
Output	SEARCH.OUT
Score	100

Trò chơi xếp bi là trò chơi hai đấu thủ và có thể mô tả như sau: Có M ô vuông xếp liên tiếp thành một hàng, các ô được đánh số thứ tự liên tiếp bắt đầu từ 1 đến M , từ trái qua phải. Trạng thái xuất phát của trò chơi được tạo ra như sau: Người ta chọn ra N ô trong số $M-1$ ô đầu tiên và thả vào các ô được chọn mỗi ô một viên bi. Hai đấu thủ luân phiên thực hiện nước đi. Đối thủ đến phiên thực hiện nước đi phải lấy một viên bi từ một ô K nào đó và đặt nó vào ô không có bi đầu tiên ở bên phải ô K . Đối thủ nào khi đến phiên của mình đưa được viên bi vào ô cuối cùng (ô số M) là người thắng cuộc.

Để khảo sát khả năng chắc thắng của đối thủ thứ nhất người ta cần giải quyết bài toán sau: "Có bao nhiêu cách bỏ N viên bi vào $M-1$ ô đầu tiên sao cho số lượng ô không chứa bi ở bên phải ô có chứa bi đầu tiên kể từ bên trái của hàng là một số lẻ.

Yêu cầu: Tính số lượng cách bỏ bi thoả mãn yêu cầu của bài toán trên

Input: Hai số nguyên N và M được ghi cách nhau bởi dấu cách ($1 \leq N < M \leq 200$)

Output: Số lượng tìm được

Example:

input	output
1 4	2

121. Hiệu quả

Program	EFFECTIVE.*
---------	-------------

Input	EFFECTIVE.INP
Output	EFFECTIVE.OUT
Score	100

Chuỗi siêu thị điện máy ABC có n cửa hàng điện máy bố trí dọc theo một con đường Quốc lộ đánh số lần lượt từ 1 đến n . Theo khảo sát, doanh thu của các cửa hàng ở vị trí i là a_i ($a_i > 0$ nếu cửa hàng bán có lãi, trường hợp ngược lại là không có lãi và lỗ vốn). Để việc kinh doanh có hiệu quả hơn, Ban Giám đốc quyết định chia dãy n cửa hàng trên thành một số dãy con gồm các cửa hàng liên tiếp nhau trên quốc lộ có tổng doanh thu không vượt quá m , mỗi dãy này sẽ thành lập một công ty con trực thuộc. Tất nhiên, Ban Giám đốc không muốn số lượng công ty con này quá nhiều. Viết chương trình tính số lượng tối thiểu các công ty con được thành lập.

Input:

- Dòng đầu tiên ghi hai số nguyên n, m ($1 \leq n \leq 10^5; m \leq 10^9$)
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$)

Dữ liệu đảm bảo luôn có cách để chia thành các công ty con theo như yêu cầu.

Output: Một số nguyên duy nhất là số lượng tối thiểu các công ty con được thành lập.

Example:

Input	Output
11 5 9 -1 2 -6 1 2 3 -4 3 9 -4	3

Subtasks:

- Subtask 1: $n \leq 5000$
- Subtask 2: $n \leq 10^5$

122. Dãy con lẻ

Program	ODDSEQ.*
Input	ODDSEQ.INP
Output	ODDSEQ.OUT
Score	100

Cho dãy số A gồm N số nguyên nằm trong phạm vi từ 1 đến N, mỗi số xuất hiện một lần. Một dãy con của A là dãy thu được từ A bằng cách xóa đi một số phần tử (có thể là 0) từ đầu và từ cuối dãy. Hãy tính xem có bao nhiêu dãy con khác nhau của A có độ dài lẻ và có phần tử trung vị đúng bằng B. Phần tử trung vị được định nghĩa là phần tử nằm ở vị trí giữa của dãy sau khi sắp xếp nó tăng dần. Ví dụ, phần tử trung vị của {5,1,3} là 3.

Input: Vào từ file văn bản ODDSEQ.INP

- Dòng đầu tiên chứa hai số nguyên N, B ($1 \leq B \leq N \leq 100000$)
- Dòng thứ hai chứa N số nguyên mô tả dãy A

Output: Vào từ file văn bản ODDSEQ.OUT một số nguyên duy nhất là số dãy con có phần tử trung vị bằng B

Example:

Input	Output
7 4 5 7 2 4 3 1 6	4

123. Sắp xếp lại

Program	PORDER.*
Input	PORDER.INP
Output	PORDER.OUT
Score	100

Cho hoán vị $P = \{p_1, p_2, \dots, p_n\}$ của $\{1, 2, \dots, n\}$. Người ta muốn sắp xếp lại hoán vị này để thu được hoán vị $\{1, 2, \dots, n\}$ bằng cách nhau sau: "Lần lượt xét các vị trí $1, 2, \dots, n$. Với mỗi vị trí i liên tục đổi chỗ p_i cho số đứng trước nó chừng nào số này lớn hơn p_i "

Ví dụ, với hoán vị $\{3, 2, 1, 5, 4\}$ qui trình đổi chỗ thực hiện như sau:

- Với $p_1 = 3$: $(3, 2, 1, 5, 4)$ 0 phép đổi chỗ
- Với $p_2 = 2$: $(3, 2, 1, 5, 4) \rightarrow (2, 3, 1, 5, 4)$ 1 phép đổi chỗ
- Với $p_3 = 1$: $(2, 3, 1, 5, 4) \rightarrow (2, 1, 3, 5, 4) \rightarrow (1, 2, 3, 5, 4)$ 2 phép đổi chỗ
- Với $p_4 = 5$: $(1, 2, 3, 5, 4)$ 0 phép đổi chỗ
- Với $p_5 = 4$: $(1, 2, 3, 5, 4) \rightarrow (1, 2, 3, 4, 5)$ 1 phép đổi chỗ

Cho biết số phép đổi chỗ của các vị trí $1, 2, \dots, n$. Hãy xác định hoán vị ban đầu

Input: Vào từ file văn bản PORDER.INP: Dòng đầu tiên ghi số nguyên dương $T \leq 5$ là số lượng tests. Tiếp theo là T nhóm dòng, mỗi nhóm mô tả một test có cấu trúc:

- Dòng 1: Ghi số nguyên dương n ($1 \leq n \leq 200000$)
- Dòng 2: Ghi các số nguyên w_1, w_2, \dots, w_n với w_i là số lần đổi chỗ mà p_i thực hiện ($i = 1, 2, \dots, n$).

Output: Ghi ra file văn bản PORDER.OUT: In ra T dòng, dòng thứ i ghi n số nguyên mô tả hoán vị p_1, p_2, \dots, p_n ban đầu ứng với test thứ i ($i = 1, 2, \dots, T$). Thứ tự test là thứ tự xuất hiện trong input.

Example:

PORDER . INP	PORDER . OUT
2	2 1 3
3	3 2 1 5 4
0 1 0	
5	
0 1 2 0 1	

Subtask:

- Subtask 1: $n \leq 10$ [30%]
- Subtask 2: $n \leq 100$ [10%]
- Subtask 3: $n \leq 500$ [10%]
- Subtask 4: $n \leq 5000$ [20%]
- Subtask 5: $n \leq 2 \cdot 10^5$ [30%]

124. Đội hình đẹp

Program	NICETEAM.*
Input	NICETEAM.INP
Output	NICETEAM.OUT
Score	100

Trong giờ Quốc phòng n bạn trong lớp đứng thành một hàng ngang đánh số từ 1 đến n , bạn thứ i có chiều cao là h_i . Cần phải chọn ra một dãy liên tiếp các bạn để tập bài đội tập đội ngũ. Một đội hình tập được gọi là đẹp nếu như:

- Tổng chiều cao của tất cả các bạn lớn hơn hoặc bằng k .
- Độ cao trung bình của các bạn được chọn là lớn nhất

Viết một chương trình tìm ra đội hình đẹp. Bạn chỉ cần tính độ cao trung bình của đội hình này làm tròn xuống thành số nguyên.

Input:

- Dòng đầu tiên ghi hai số nguyên dương n và k ($1 \leq n \leq 10^6; 0 \leq k \leq 10^9$)
- Dòng thứ hai ghi lần lượt các số nguyên dương h_1, h_2, \dots, h_i ($h_i \leq 10^9$)

Output: Một số nguyên duy nhất là độ cao trung bình của đội hình đẹp tìm được, làm tròn xuống thành số nguyên

Example:

Input	Output
5 6 1 5 2 4 3	3

Subtasks:

- Subtask 1: $n \leq 5000$
- Subtask 2: $n \leq 10^5$

125. Giá trị biểu thức ngoặc

Program	PAREN.*
Input	PAREN.INP
Output	PAREN.OUT
Score	100

Cho dãy ngoặc đúng (chỉ có dấu mở ngoặc '(' và dấu đóng ngoặc ')'). Người ta tính giá trị của một dãy ngoặc đúng theo qui tắc sau:

1. Giá trị của () là 1
2. Nếu A có giá trị x thì (A) có giá trị 2x
3. Nếu A, B có giá trị x, y thì AB có giá trị x+y

Ví dụ giá trị của '(()())' là $2*1+1=3$

Cho một biểu thức ngoặc đúng độ dài N ($2 \leq N \leq 10^5$). Hãy xác định giá trị của nó

Input:

- Dòng đầu tiên ghi số nguyên dương N
- N dòng tiếp theo, dòng thứ i ghi số 0 nếu vị trí tương ứng là '(' và ghi số 1 nếu là ')'

Output: Một số nguyên duy nhất là kết quả tìm được. Vì số này có thể rất lớn nên chỉ cần lấy phần dư của nó khi chia cho 12345678910.

Example:

input	output
6 0 0 1 1 0 1	3

126. Sửa hàng rào

Program	WALL.*
Input	WALL.INP
Output	WALL.OUT
Score	100

Bờm dựng đã dựng xong một hàng rào gồm n thanh gỗ độ rộng đơn vị ghép liền nhau, đánh số từ 1 đến n , thanh gỗ thứ i có độ cao là a_i . Tuy vậy, vẫn còn thừa m thanh gỗ nữa chưa sử dụng. Anh ta quyết định dùng m thanh gỗ còn thừa này để nâng chiều cao của hàng rào vừa dựng xong. m thanh gỗ này được Bờm xếp lên một chiếc xe thành một chồng, tính từ trên xuống dưới các thanh gỗ được đánh số từ 1 đến m , thanh gỗ thứ i có chiều cao là b_i ($i = 1, 2, \dots, m$). Bờm sẽ kéo chiếc xe này từ đầu hàng rào đến cuối hàng rào (từ thanh số 1 đến thanh số n). Mỗi khi đến vị trí của một thanh gỗ anh ta có thể:

- Hoặc bỏ qua không làm gì cả
- Hoặc lấy một thanh gỗ bất kỳ còn lại trên xe ba gác (khi đó tất cả các thanh gỗ ở bên trên nó sẽ bị bỏ ra khỏi xe ba gác và không bao giờ sử dụng nữa) đóng tiếp lên vị trí của thanh gỗ

đang đứng, chiều cao mới tại vị trí này sẽ là tổng chiều cao của thanh gỗ cũ và thanh gỗ mới đóng lên.

Hãy xác định độ lớn nhất của hàng rào sau khi nâng chiều cao theo cách nói trên. Độ cao của hàng rào được tính là độ cao của thanh gỗ có độ cao thấp nhất.

Input: Gồm nhiều test ghi liên tiếp nhau. Mỗi test có cấu trúc như sau:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$)
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^8, i = 1, 2, \dots, n$)
- Dòng thứ ba chứa số nguyên m ($1 \leq m \leq 10^5$)
- Dòng thứ ba chứa m số nguyên b_1, b_2, \dots, b_m ($1 \leq b_i \leq 10^5, i = 1, 2, \dots, m$)

Output: Kết quả các test ghi lần lượt theo cấu trúc:

- Dòng đầu ghi hai số nguyên h và k - độ cao lớn nhất của hàng rào và số thanh gỗ được đóng thêm.
- Mỗi dòng trong k dòng tiếp theo ghi hai số nguyên u và v trong đó u - số hiệu của thanh gỗ được đóng cao hơn và v - số hiệu của thanh gỗ thừa được dùng để đóng.

Example:

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

127. Đoạn thẳng cắt nhau

Program INTERSECT.*
 Input INTERSECT.INP
 Output INTERSECT.OUT
 Score 100

Trên mặt phẳng tọa độ cho hai đoạn thẳng. Hỏi rằng hai đoạn thẳng có cắt nhau hay không?

Input:

- Dòng đầu ghi số nguyên T - số bộ dữ liệu ($T \leq 10$)
- T dòng tiếp theo, mỗi dòng mô tả một bộ dữ liệu gồm 8 số nguyên cách nhau bởi dấu cách $xa, ya, xb, yb, xc, yc, xd, yd$ thể hiện hai đoạn thẳng nối điểm $A(xa, ya)$ với $B(xb, yb)$ và nối điểm $C(xc, yc)$ với điểm $D(xd, yd)$

Output: Gồm T dòng, ghi YES/NO tùy theo bộ test tương ứng cho kết quả là cắt nhau/không cắt nhau.

Example:

input	output
2	YES
0 0 2 0 1 1 1 -1	NO
0 0 2 0 1 1 1 4	

128. Đa giác

Program POLY.*
 Input POLY.INP
 Output POLY.OUT
 Score 100

Trên mặt phẳng tọa độ, xét đa giác lồi n đỉnh, các đỉnh đều có tọa độ nguyên và có giá trị tuyệt đối không vượt quá 10^5 . Các đỉnh của đa giác được liệt kê theo chiều kim đồng hồ.

Yêu cầu: Cho đoạn thẳng xác định bởi hai điểm có tọa độ là (x_1, y_1) và (x_2, y_2) trong đó x_1, y_1, x_2, y_2 là các số nguyên và có giá trị tuyệt đối không vượt quá 10^5 . Hãy xác định độ dài L là phần của

đoạn thẳng nằm trong đa giác hay trên cạnh của đa giác và đưa ra số nguyên là phần nguyên của tích ($L * 100$).

Input:

- Dòng đầu tiên chứa số nguyên n ($3 \leq n \leq 100$)
- Dòng thứ i trong n dòng sau chứa 2 số nguyên xác định tọa độ đỉnh i của đa giác,
- Dòng cuối cùng chứa 4 số nguyên x_1, y_1, x_2, y_2 .

Hai số liên tiếp trên một dòng cách nhau một dấu cách.

Output: Ghi một số nguyên là phần nguyên của tích ($L * 100$).

Example:

input	output
0 1 1 0 0 -1 -1 0 -2 0 0 0	100

129. Bao lồi

Program CONVEX.*
Input CONVEX.INP
Output CONVEX.OUT
Score 100

Cho n điểm có tọa độ nguyên trên mặt phẳng ($3 \leq n \leq 2 \cdot 10^5$). Tọa độ của mỗi điểm có trị tuyệt đối không vượt quá 10^9 . Có thể có những điểm trùng nhau, tuy vậy bao giờ cũng tìm được ít nhất ba điểm không thẳng hàng.

Yêu cầu: Xác định đa giác lồi có diện tích nhỏ nhất chứa tất cả các đỉnh đã cho (tính cả trên biên) và không có ba điểm nào thẳng hàng. Chỉ ra các đỉnh của đa giác theo trình tự duyệt ngược chiều kim đồng hồ, tính chu vi và diện tích của đa giác.

Input:

- Dòng đầu tiên chứa số nguyên dương n
- n dòng tiếp theo, mỗi dòng ghi hai số nguyên - tọa độ của một điểm.

Output:

- Dòng đầu tiên ghi số nguyên dương - số đỉnh của đa giác lồi
- Dòng thứ hai chứa số thực - chu vi của đa giác lồi với 2 chữ số phần thập phân
- Dòng thứ ba chứa số thực - diện tích của đa giác lồi với một chữ số phần thập phân

Example:

Input	Output
9 0 0 1 1 2 2 1 0 0 1 2 0 0 2 2 1 1 2	4 8.00 4.0

130. Bộ ba điểm thẳng hàng

Program TRIPOINT.*
Input TRIPOINT.INP
Output TRIPOINT.OUT
Score 100

Trên mặt phẳng tọa độ cho N điểm. Hỏi rằng có bao nhiêu bộ ba điểm thẳng hàng (cùng nằm trên một đường thẳng)

Input:

- Dòng đầu tiên ghi số nguyên dương $N \leq 2000$
- N dòng tiếp theo, mỗi dòng ghi tọa độ của một điểm. Tọa độ các điểm có trị tuyệt đối không vượt quá 10000

Output: Một số nguyên duy nhất là số lượng bộ ba điểm thẳng hàng

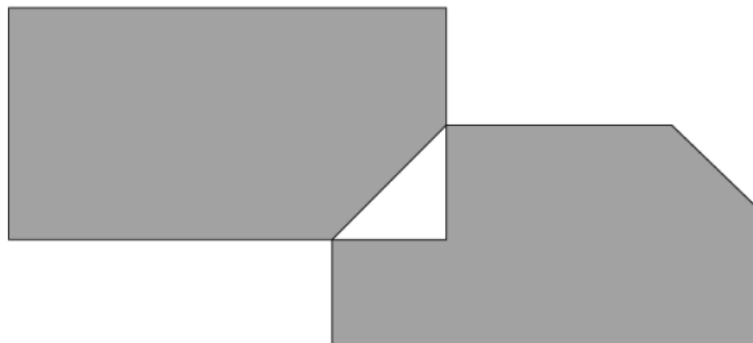
Example:

input	output
6 0 0 0 1 0 2 1 1 1 2 2 2	3

131. Giao hai đa giác

Program POLYGONS.*
Input POLYGONS.INP
Output POLYGONS.OUT
Score 100

Cho 2 đa giác lồi. Chúng có thể chồng hoặc không chồng lên nhau. Viết chương trình đọc tọa độ các đỉnh của hai đa giác lồi và tính toán phần diện tích độc nhất của hai đa giác, tức là phần diện tích được giới hạn chính xác bởi một trong hai đa giác. Ví dụ trong hình sau phần diện tích cần tìm là phần được in đậm:



Input: Chứa nhiều test, mỗi test chứa một cặp, mỗi cặp gồm 2 dòng, một dòng đại diện cho một đa giác. Mỗi dòng sẽ chứa số đỉnh của đa giác không vượt quá 100, tiếp theo đó là các cặp số nguyên biểu diễn cặp tọa độ x, y của các đỉnh theo chiều kim đồng hồ. Tất cả tọa độ là các số nguyên dương nhỏ hoặc bằng 100.

Output: Với mỗi cặp của 2 đa giác chương trình cần in ra phần diện tích tính được với 2 chữ số phần thập phân, các kết quả in cách nhau một dấu cách.

Example:

input	output
3 5 5 8 1 2 3 3 5 5 8 1 2 3 4 1 2 1 4 5 4 5 2 6 6 3 8 2 8 1 4 1 4 2 5 3 0	0.00 13.50

132. Điểm Fecmat-Torricelli

Program	FTPOINT.*
Input	FTPOINT.INP
Output	FTPOINT.OUT
Score	100

Trong hình học phẳng, điểm Fecmat của một tam giác cũng còn được gọi là điểm Torricelli hoặc Fecmat-Torricelli là một điểm sao cho tổng khoảng cách từ điểm đó đến các đỉnh của tam giác là bé nhất. Trong bài toán này ta sẽ mở rộng khái niệm điểm Torricelli cho n điểm:

Trên mặt phẳng tọa độ cho n điểm có tọa độ lần lượt là $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Hãy tìm điểm nằm trong hình chữ nhật $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ sao cho tổng khoảng cách đến n điểm đã cho là nhỏ nhất. Ở đây x_{\max} , x_{\min} là giá trị lớn nhất và nhỏ nhất của các hoành độ còn y_{\max} , y_{\min} là giá trị lớn nhất và nhỏ nhất của các tung độ/

Input:

- Dòng đầu tiên chứa số nguyên dương $n \leq 10^3$
- n dòng sau, dòng thứ i ghi hai số nguyên x_i, y_i là tọa độ của điểm thứ i ($|x_i|, |y_i| \leq 10^6$)

Output: Một số thực duy nhất là tổng khoảng cách nhỏ nhất tìm được với 3 chữ số phần thập phân

Example:

input	output
3 0 0 3 3 3 4	5.243

Ghi chú: Có 50% số test tất cả các điểm nằm trên một đường thẳng

133. Luồng cực đại

Program	MAXFLOW.*
Input	MAXFLOW.INP
Output	MAXFLOW.OUT
Score	100

Cho mạng có N đỉnh. Hãy tìm luồng cực đại với đỉnh phát N và đỉnh thu 1

Input:

- Dòng thứ nhất ghi số $N \leq 5000$
- Tiếp theo là một số dòng (không quá 10000 dòng) mỗi dòng mô tả một cung (u, v) gồm có 3 số u, v, w thể hiện cung nối từ u đến v có độ thông qua w

Output: Một số nguyên duy nhất là luồng cực đại tìm được

Example:

Input	Output
7 3 1 7 6 1 47 7 2 12 7 2 20 2 4 11 7 3 15 5 3 20 7 4 10 4 2 21 4 6 16 6 5 23	23

134. Ghép nhanh!

Program	FMATCH.*
Input	FMATCH.INP

Output FMATCH.OUT
Score 100

Cho đồ thị hai phía $G = (X, Y)$ với $m = |X|, n = |Y|$. Hãy tìm cặp ghép có số cạnh nhiều nhất trên G .

Input:

- Dòng đầu tiên ghi ba số nguyên m, n, p ($1 \leq m, n \leq 50000; 1 \leq p \leq 150000$)
- p dòng tiếp theo mỗi dòng ghi hai số nguyên u, v ($1 \leq u \leq m; 1 \leq v \leq n$) mô tả một cạnh nối

Output: Một số nguyên duy nhất là số cạnh của bộ ghép cực đại

Example:

Input	Output
5 4 6 5 2 1 2 4 3 3 1 2 2 4 4	3

135. Năng suất dây chuyền

Program NANGSUAT.*
Input NANGSUAT.INP
Output NANGSUAT.OUT
Score 100

Một dây chuyền sản xuất có n vị trí làm việc đánh số từ 1 đến n . Có n công nhân để xếp vào làm việc trên các vị trí này. Biết s_{ij} là năng suất để công nhân i trên vị trí làm việc j của dây chuyền ($i, j=1, 2, \dots, n$). Cho trước một cách bố trí công nhân đứng làm việc trên các vị trí của dây chuyền, ta có thể tính năng suất của dây chuyền theo cách bố trí đã cho như là năng suất nhỏ nhất của công nhân trên dây chuyền. *Yêu cầu:* Tìm cách bố trí N công nhân vào làm việc trên N vị trí của một dây chuyền sản xuất sao cho năng suất của dây chuyền là lớn nhất.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($n \leq 200$)
- Dòng thứ i trong số n dòng tiếp theo chứa n số nguyên dương $s_{i1}, s_{i2}, \dots, s_{in}$ ($i = 1, 2, \dots, n$). Các số này có giá trị không vượt quá 20000 và cách nhau bởi dấu trống.

Output:

- Dòng đầu tiên ghi năng suất của dây chuyền theo cách bố trí tìm được.
- Dòng thứ i trong số n dòng tiếp theo ghi vị trí làm việc của công nhân i trên dây chuyền theo cách bố trí tìm được.

Example:

Input	Output
4 9 4 4 12 8 7 8 13 2 2 8 3 6 7 3 7	7 1 4 3 2

136. Chọn vận động viên

Program ATHLETE.*
Input ATHLETE.INP
Output ATHLETE.OUT
Score 100

Kết quả thi đấu quốc gia của n vận động viên (đánh số từ 1 đến n) trên m môn (đánh số từ 1 đến m) được đánh giá bằng điểm (giá trị nguyên không âm). Với mỗi vận động viên ta biết điểm đánh giá trên từng môn của vận động viên ấy.

Cần chọn ra k vận động viên và k môn ($1 \leq k \leq \min(m, n)$) để thành lập đội tuyển thi đấu Olympic quốc tế, trong đó mỗi vận động viên chỉ được thi đấu đúng một môn sao cho tổng số điểm của các vận động viên trên các môn đã chọn là lớn nhất.

Input:

- Dòng đầu ghi hai số n, m ($1 \leq n, m \leq 200$)
- n dòng tiếp theo, mỗi dòng ghi các điểm đánh giá trên tất cả các môn theo thứ tự môn thi 1, 2, ..., m . Các dòng này được ghi theo thứ tự vận động viên 1, 2, ..., n . Các số trên cùng dòng ghi cách nhau dấu cách. Điểm nằm trong phạm vi từ 0 đến 1000
- Dòng cuối cùng ghi số k

Output:

- Dòng đầu ghi tổng số điểm của đội tuyển Olympic
- Trong k dòng tiếp theo mỗi dòng ghi hai số u, v thể hiện vận động viên u sẽ được cử thi đấu môn v .

Example:

Input	Output
3 3	11
1 5 0	2 1
5 7 4	3 2
3 6 3	
2	

137. Đảo tham lam

Program GREED.*
 Input GREED.INP
 Output GREED.OUT
 Score 100

Trên đường đi tìm cha, cậu bé Gon lạc đến đảo Tham Lam. Hòn đảo này rất kỳ quái, người dân không dùng tiền mà dùng thẻ để trao đổi. Trên mỗi thẻ ghi một số nguyên nằm trong khoảng $[1, N]$ và được gọi là mã số của thẻ. Chỉ có một cách duy nhất để ra khỏi đảo là đem được N thẻ có mã đôi một khác nhau (từ là mã $1, 2, \dots, N$) đổi lấy vé tàu.

Gon có 2 cách để kiếm thẻ ở trên đảo:

- Nhặt thẻ mà người khác đánh rơi
- Trao đổi với ngân hàng của đảo: dùng 1 thẻ của mình đổi lấy 1 thẻ khác của ngân hàng, lệ phí 1 lần đổi là 1 cục vàng (ngân hàng dùng để đúc thẻ mới)

Rất may là Gon được một người bạn tốt bụng tặng cho N thẻ nên chỉ còn phải ra ngân hàng đổi thẻ (để được N thẻ khác nhau đôi một). Vì chuyến đi dài nên Gon phải tiết kiệm vàng. Bạn hãy giúp Gon tìm cách đổi để tốn ít vàng nhất mà vẫn đổi được vé tàu đi ra khỏi đảo. Biết rằng luôn tồn tại ít nhất một cách đổi.

Input:

- Dòng thứ nhất ghi số nguyên N ($N \leq 100$)
- Dòng thứ hai ghi N số nguyên là mã số của N thẻ mà Gon có
- Tiếp theo là một số dòng, trên mỗi dòng chứa hai số u, v có nghĩa là có thể đổi thẻ có mã số u của Gon lấy thẻ có mã số v của ngân hàng và ngược lại.

Các dữ liệu số trên cùng một dòng được ghi cách nhau bởi ít nhất một dấu cách.

Output: Một số nguyên duy nhất là tổng số vàng ít nhất phải trả khi đổi thẻ

Example:

Input	Output
-------	--------

4	4
1 1 1 1	
1 2	
2 3	
1 4	
3 4	

138. Chiếu sáng

Program	LIGHTING.*
Input	LIGHTING.INP
Output	LIGHTING.OUT
Score	100

Bản đồ công viên Disneyland là một hình chữ nhật kích thước $m \times n$ được chia thành các lưới ô vuông đơn vị m hàng, n cột. Các hàng của lưới được đánh số từ 1 đến m , các cột của lưới được đánh số từ 1 đến n . Ô nằm ở hàng i , cột j được gọi là (i, j) . Người ta đặt k khu vui chơi vào một số ô của lưới, mỗi khu vui chơi nằm hoàn toàn trong một ô và có thể có nhiều khu vui chơi nằm trong cùng một ô.

Hệ thống chiếu sáng của công viên gồm m đèn loại A : a_1, a_2, \dots, a_m và n đèn loại B : b_1, b_2, \dots, b_n . Đèn a_i có thể chiếu sáng tất cả các ô trên hàng i và đèn b_j có thể chiếu sáng tất cả các ô nằm trên cột j . ($1 \leq i \leq m; 1 \leq j \leq n$)

Yêu cầu: Hãy bật sáng một số ít nhất các đèn để chiếu sáng toàn bộ các ô có khu vui chơi.

Input:

- Dòng 1 chứa ba số nguyên dương $m, n, k \leq 10^5$
- k dòng tiếp theo, dòng thứ i ghi chỉ số hàng và chỉ số cột của khu vui chơi thứ i .

Output: Một số nguyên duy nhất là tổng lượng đèn ít nhất cần bật.

Example:

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

	1	2	3	4	5
1					
2					
3					
4					
5					

139. Các cuộc thám hiểm không gian

Program	NASA.*
Input	NASA.INP
Output	NASA.OUT
Score	100

Giáo sư Spock đang cố vấn cho NASA, cơ quan này đang hoạch định một loạt các chuyến bay con thoi không gian và phải quyết định các cuộc thí nghiệm thương mại để thực hiện và các dụng cụ phải có trên tàu cho mỗi chuyến bay. Với mỗi chuyến bay, NASA xét một tập hợp các thí nghiệm có thể thực hiện, các thí nghiệm này đánh số từ 1 đến N , với thí nghiệm j , các nhà bảo trợ thương mại đã đồng ý thanh toán cho NASA p_j đôla ($j=1,2,\dots,N$). Các cuộc thí nghiệm sử dụng M dụng cụ, đánh số từ 1 đến M . Mỗi thí nghiệm sẽ sử dụng một số trong số M dụng cụ này. Mức hao phí để mang dụng cụ thứ k vào không gian là c_k đôla ($k=1,2,\dots,M$). Công việc của giáo sư Spock là xác định các cuộc thí nghiệm để thực hiện và các dụng cụ để mang theo cho một chuyến bay đã cho sao cho tối đa

hoá lợi tức ròng, tức là tổng thu nhập từ các cuộc thí nghiệm được thực hiện trừ đi mức hao phí của tất cả các dụng cụ mang theo.

Yêu cầu: Viết chương trình giúp giáo sư Spock thực hiện yêu cầu trên.

Input:

- Dòng đầu tiên chứa hai số nguyên N, M ($1 \leq N, M \leq 100$)
- Dòng thứ hai chứa N số nguyên p_1, p_2, \dots, p_n
- Dòng thứ ba chứa M số nguyên c_1, c_2, \dots, c_m
- Tiếp theo là N dòng, dòng thứ i chứa các số $a_{i1}, a_{i2}, \dots, a_{im}$ với a_{ij} bằng 1 nếu thí nghiệm i sử dụng dụng cụ j và bằng 0 nếu thí nghiệm i không sử dụng dụng cụ j .

Output:

- Dòng đầu tiên ghi tổng lợi tức ròng thu được.
- Dòng 2 liệt kê danh sách các thí nghiệm cần thực hiện (nếu không có thí nghiệm nào thực hiện thì ghi số 0)
- Dòng 3 liệt kê danh sách các dụng cụ mà chuyến bay phải mang theo (nếu không có dụng cụ nào phải mang theo thì ghi 0)

Example:

Input	Output
3 4	16
4 10 11	2 3
6 2 3 7	2 3
1 0 0 1	
0 1 1 0	
0 1 0 0	

140. Phủ sóng đô thị

Program COVERAGE.*
 Input COVERAGE.INP
 Output COVERAGE.OUT
 Score 100

Tại một thành phố có N hộ dân cư và 2 nhà cung cấp dịch vụ viễn thông. Với mỗi hộ dân cư i , nếu họ chọn nhà cung cấp thứ nhất sẽ mất chi phí là a_i , nếu họ chọn nhà cung cấp thứ hai sẽ mất chi phí là b_i . Ngoài ra, nếu 2 hộ dân cư i và j lắp dịch vụ của 2 nhà cung cấp khác nhau, họ phải tốn thêm một chi phí là c_{ij} cho việc lắp đặt hệ thống chống nhiễu sóng.

Yêu cầu: Hãy tính chi phí nhỏ nhất để phủ sóng toàn bộ N hộ dân cư.

Input:

- Dòng đầu ghi số N . ($1 \leq N \leq 200$)
- Dòng thứ 2 ghi N số a_i . ($1 \leq a_i \leq 1000$)
- Dòng thứ 3 ghi N số b_i . ($1 \leq b_i \leq 1000$)
- N dòng cuối thể hiện c_{ij} . ($0 \leq c_{ij} = c_{ji} \leq 100$)

Output: Một số duy nhất thể hiện chi phí nhỏ nhất.

Example:

Input	Output
3	5
1 1 10	
10 10 1	
0 0 1	
0 0 1	
1 1 0	

141. Phá hủy đô thị

Program DESTROY.*
 Input DESTROY.INP

Output DESTOYR.OUT
Score 100

Cho một đồ thị có hướng, với mỗi đỉnh v thuộc đồ thị, bạn có thể thực hiện các phép sau đây:

- Xóa tất cả các cung xuất phát từ v , tốn a_v đơn vị.
- Xóa tất cả các cung kết thúc ở v , tốn b_v đơn vị.

Hãy tìm cách xóa tất cả các cung của đồ thị với chi phí nhỏ nhất có thể.

Input:

- Dòng đầu ghi 2 số N, M lần lượt là số đỉnh và số cạnh của đồ thị. ($1 \leq N \leq 200$)
- Dòng thứ 2 ghi N số a_v .
- Dòng thứ 3 ghi N số b_v .
- M dòng tiếp theo, mỗi dòng ghi một cặp số u, v thể hiện 1 cung của đồ thị, xuất phát từ u và kết thúc ở v .

Output: Ghi chi phí nhỏ nhất để xóa tất cả các cung của đồ thị.

Example:

Input	Output
4 4 1 1 1 1 1 1 1 1 1 2 2 4 1 3 3 4	2

142. Điều tra

Program INQUIR.*
Input INQUIR.INP
Output INQUIR.OUT
Score 100

Hãng zZz có hệ thống N siêu thị, đánh số $1, 2, \dots, N$. Gần đây hãng phát hiện K vụ trộm đồ ở các siêu thị. Hãng nghi vấn các vụ trộm này có cùng hung thủ nên muốn nhờ bạn dựa vào thông tin địa điểm và thời gian xảy ra các vụ trộm, thời gian di chuyển trên các đường nối trực tiếp giữa các siêu thị, xác định xem có ít nhất bao nhiêu hung thủ mới có thể thực hiện tất cả các vụ trộm.

Input:

- Dòng 1: số nguyên T ($1 \leq T \leq 100$) là số tests
- Mỗi test được cho trên một nhóm dòng, bao gồm:
- Dòng 1: ba số nguyên N, K, M ($1 \leq N, K \leq 500; 1 \leq M \leq 10^5$) tương ứng là số siêu thị, số vụ trộm, số đường nối trực tiếp các cặp siêu thị
- Dòng 2... $K+1$: mỗi dòng ghi thông tin của một vụ trộm, gồm hai số nguyên t, v ($t \leq 10^6$) chỉ vụ trộm xảy ra ở thời điểm t tại siêu thị v
- Dòng $K+2$... $K+M+1$: mỗi dòng ba số nguyên a, b, c ($1 \leq c \leq 10^6$) chỉ một đường hai chiều nối trực tiếp hai siêu thị a, b tốn thời gian di chuyển c .

Output: gồm T dòng, dòng i ghi số nguyên kết quả của test thứ i .

Example:

Input	Output
2 2 3 1 10 1 18 2 20 1 1 2 5	2 1

2 3 2	
10 1	
18 2	
20 1	
1 2 5	
1 2 1	

143. Tam giác vuông

Program SQTRI.*
 Input SQTRI.INP
 Output SQTRI.OUT
 Score 100

Cho N điểm trên mặt phẳng tọa độ.

Viết chương trình xác định xem có bao nhiêu tam giác vuông với 3 đỉnh là 3 điểm trong số N điểm nói trên.

Input:

- Dòng đầu tiên ghi số nguyên dương N ($N \leq 1500$)
- N dòng sau, mỗi dòng ghi tọa độ một điểm. Tọa độ là số nguyên nằm trong khoảng $[-10^9, 10^9]$.

Output: Số nguyên duy nhất là số lượng tam giác vuông tìm được

Example:

input	output
5 -1 1 -1 0 0 0 1 0 1 1	7

144. Thiệp xạ

Program ARCHER.*
 Input ARCHER.INP
 Output ARCHER.OUT
 Score 100

Dân chúng vùng Nottinghamshire đề nghị Robin Hood biểu diễn bắn cung. Buổi biểu diễn được thực hiện trong không gian với hệ trục tọa độ Descartes vuông góc Oxyz, trong đó mặt phẳng Oxy (bao gồm các điểm có tọa độ $z = 0$) là mặt đất. Trục Oz hướng lên trên. Người ta thả n quả bóng bay đánh số từ 1 đến n , quả bóng thứ i có tọa độ (x_i, y_i, z_i) trong đó $z_i > 0$ (có thể có nhiều quả bóng cùng một tọa độ). Robin Hood cần phải chọn một vị trí A trên mặt đất và bắn một mũi tên đi thẳng theo một tia gốc A, mũi tên sẽ bắn xuyên qua tất cả các quả bóng nằm trên tia đó.

Yêu cầu: Hãy giúp Robin Hood xác định nhiều nhất số quả bóng có thể bắn trúng.

Input:

- Dòng 1 chứa số nguyên dương $n \leq 2000$
- n dòng tiếp theo, dòng thứ i chứa ba số nguyên x_i, y_i, z_i có trị tuyệt đối không vượt quá 2000, $z_i > 0$

Output: Một số nguyên duy nhất là số lượng nhiều nhất các quả bóng có thể bắn trúng

Example:

input	output
-------	--------

- Trong N dòng tiếp theo, dòng thứ i ghi hai số nguyên x_i, y_i lần lượt là hoành độ và tung độ của điểm thứ i trong số N đỉnh đa giác (các đỉnh của đa giác được liệt kê theo chiều xuôi hoặc ngược kim đồng hồ)

Output: Một số thực x_0 với ý nghĩa đường ranh giới dùng để chia đất là đường thẳng $x=x_0$ (x_0 viết với 4 chữ số phân thập phân)

Example:

input	output
4 0 0 2 0 2 2 0 2	1.0000

147. Điểm trong tam giác

Program TRIANGLE.*
Input TRIANGLE.INP
Output TRIANGLE.OUT
Score 100

Trong mặt phẳng tọa độ Oxy cho N điểm A_1, A_2, \dots, A_N và điểm P . Hãy xác định điểm P hoàn toàn nằm trong bao nhiêu tam giác $A_i A_j A_k$.

Input:

- Dòng 1: số nguyên N ($3 \leq N \leq 10^5$)
- Dòng 2: hai số nguyên là tọa độ điểm P
- Dòng 3 ... $N + 2$: dòng $i + 2$ chứa hai số nguyên là tọa độ điểm A_i

Dữ liệu đảm bảo các điểm là đôi một phân biệt, các tọa độ thuộc phạm vi $-10^9 \dots 10^9$.

Output: một số nguyên là kết quả tìm được

Example:

TRIANGLE . INP	TRIANGLE . OUT
3 0 0 0 1 -1 -1 1 -1	1

Subtasks:

- Subtask 1: $N \leq 500$ [30%]
- Subtask 2: $N \leq 2000$ [30%]
- Subtask 3: $N \leq 10^5$ [40%]

148. Hang động

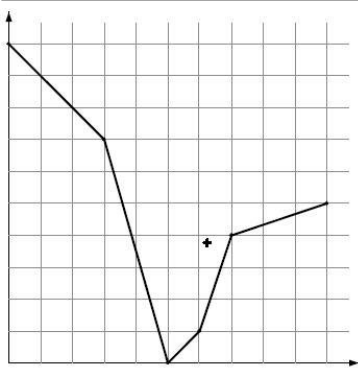
Program SPILJA.*
Input SPILJA.INP
Output SPILJA.OUT
Score 100

Gần ngôi làng có một hang động tổ tiên của Mirko đã sống hàng nghìn năm về trước. Muốn là người đầu tiên phát hiện ra những di vật cổ đại này, Mirko chuẩn bị cho chuyến đi khám phá hang động. Do hang động không có điện nên Mirko phải mua một ngọn đèn. Mirko muốn chọn một vị trí để từ đó có thể nhìn thấy toàn bộ nền hang động.

Tưởng tượng rằng nền hang động là một đường gấp khúc trong hệ tọa độ gồm N đỉnh t_1, t_2, \dots, t_N .

Nền hang động luôn chạy từ trái sang phải, nghĩa là với mọi $i=1, 2, \dots, N-1$ tọa độ x của t_i luôn bé hơn tọa độ x của t_{i+1} .

Ví dụ: (một lời giải cho ví dụ dưới đây)



Ngon đèn phải đặt ở một điểm nào đó "phía trên" nền hang động sao cho chiếu sáng được toàn bộ nền hang động. Chính xác hơn, tọa độ x của ngọn đèn phải được đặt giữa tọa độ x của điểm đầu và điểm cuối của hang động, và tọa độ y của ngọn đèn phải lớn hơn hoặc bằng tọa độ y của điểm nằm trên nền hang động ở cùng tọa độ x.

Ngon đèn chiếu sáng toàn bộ hang động nếu với mọi điểm thuộc nền hang động, đoạn thẳng nối điểm đó và ngọn đèn không cắt đường gấp khúc thể hiện nền hang động. Tuy nhiên, đoạn thẳng và đường gấp khúc có thể giao nhau tại các đỉnh hoặc dọc theo một đoạn thẳng thuộc

đường gấp khúc.

Yêu cầu: hãy giúp Mirko xác định độ cao nhỏ nhất có thể đặt ngọn đèn để chiếu sáng toàn bộ nền hang động.

Input:

- Dòng đầu tiên chứa số nguyên N ($2 \leq N \leq 50000$) là số đỉnh của nền hang động.
- Dòng thứ i trong N dòng tiếp theo chứa 2 số nguyên X_i, Y_i ($0 \leq X_i, Y_i \leq 100000$), là tọa độ đỉnh thứ i của nền hang động. Các giá trị X_i có thứ tự tăng dần.

Output: In ra 1 số thực duy nhất là tọa độ y bé nhất có thể đặt được ngọn đèn, với độ chính xác 2 chữ số thập phân.

Example:

Input	Output
6 0 10 3 7 5 0 6 1 7 4 10 5	3.75

149. Phân tích

Program	SUMS.*
Input	SUMS.INP
Output	SUMS.OUT
Score	100

Cho n số nguyên dương a_1, a_2, \dots, a_n và số nguyên dương S . Hãy đếm xem có bao nhiêu cách phân tích S thành tổng các số nguyên dương trong n số ở trên. Mỗi số sử dụng không quá một lần

Input:

- Dòng đầu tiên ghi hai số nguyên dương n và S ($n \leq 40, S \leq 10^{18}$)
- Dòng thứ hai ghi n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^{18}$)

Output: Một số nguyên duy nhất là số cách phân tích S thành tổng các số trong n số a_1, a_2, \dots, a_n mỗi số sử dụng không quá một lần

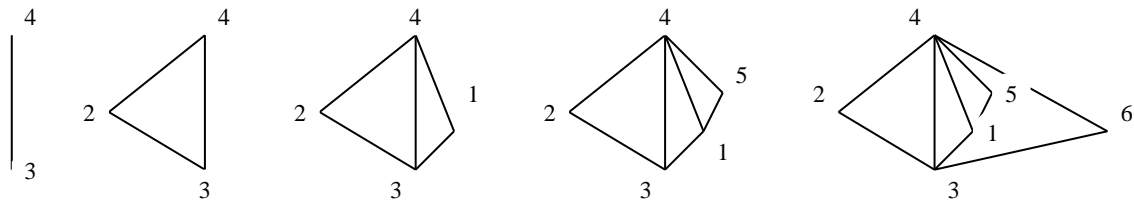
Example:

input	output
4 6 1 2 3 4	2

150. Hành tinh X

Program	XPLANET.*
Input	XPLANET.INP
Output	XPLANET.OUT
Score	100

Mọi người dân trong hành tinh X đều dựng nhà hình tam giác. Để tránh thủ thời gian, cách xây dựng của họ rất đặc biệt. Đầu tiên họ xây một bức tường thẳng, sau đó, họ chỉ việc xây thêm hai bức tường mới ghép với một bức tường đã có sẵn, kết quả được một ngôi nhà kín hình tam giác. Tất nhiên, hai bức tường mới được xây thêm vẫn có thể dùng để bắt đầu xây thêm các nhà mới. Có khi một số nhà nằm trong một số nhà khác (xem hình vẽ trong đó thể hiện quá trình xây một số ngôi nhà).



Để chiếu sáng các nhà, họ đặt ở mỗi góc nhà một bóng đèn, các nhà chung góc có thể dùng chung đèn đặt tại góc đó. Tại mỗi góc có một công tắc chuyển đổi trạng thái (tắt \rightarrow bật, bật \rightarrow tắt) không những bóng đèn ở góc đó mà còn các bóng đèn ở các góc cùng tường với góc đó.

Cho trước tình trạng của hệ các bóng đèn, hãy tìm một dãy bấm các công tắc sao cho mọi đèn tại mọi góc đều sáng và số lượng công tắc được bấm là ít nhất có thể được.

Input: trong đó dòng thứ nhất ghi số N là số góc nhà, $3 \leq N \leq 200$, các góc nhà đánh số từ 1 đến N . Tiếp theo là $2N-3$ dòng, mỗi dòng ghi hai số i, j với ý nghĩa có tường nối hai góc i và j . Dòng cuối cùng ghi N số mà số thứ i bằng 1 hoặc 0 tương ứng với đèn ở góc i sáng hoặc tối, dãy này thể hiện tình trạng cho trước của dãy các bóng đèn tại các góc. Dữ liệu phù hợp với mô tả bài toán.

Output: Dòng thứ nhất ghi số nguyên K mà nếu bài toán không có lời giải, $K=-1$, nếu có lời giải, K là số các công tắc cần bấm, tiếp theo là K dòng, mỗi dòng ghi một số hiệu của công tắc được bấm.

Example:

Input	Output
6	2
1 3	1
1 4	6
1 5	
2 3	
2 4	
3 4	
3 6	
4 5	
4 6	
0 1 1 1 0 0	

151. Khu vui chơi

Program	LAND.*
Input	LAND.INP
Output	LAND.OUT
Score	100

Một khu đất kích thước $m \times n$. Theo qui hoạch ban đầu trên khu đất đó sẽ xây dựng k khu vui chơi, khu vui chơi thứ i sẽ là một khu vực với kích thước $a_i \times b_i$. Tuy nhiên một vấn đề mới phát sinh, người ta muốn qui hoạch lại để có thêm một khu vui chơi có dạng hình vuông và hình vuông diện tích càng lớn càng tốt. Các khu vui chơi được xây dựng có thể tiếp xúc với nhau nhưng không được giao nhau.

Yêu cầu: Cho m, n và k khu vui chơi, hãy tính diện tích của khu đất hình vuông lớn nhất tìm được.

Input:

- Dòng đầu ghi ba số nguyên m, n, k ($m, n \leq 10, k \leq 5$)
- k dòng sau, mỗi dòng ghi hai số nguyên a_i, b_i

Output: Gồm một dòng ghi một số nguyên là diện tích của khu đất hình vuông lớn nhất tìm được.

Example:

input	output
5 5 2 1 5 1 4	16

152. Lát gạch

Program BRICK.*
Input BRICK.INP
Output BRICK.OUT
Score 100

Cho một hình chữ nhật kích thước $2 \times n$ ($n \leq 10^9$). Hãy đếm số cách lát các viên gạch có kích thước 1×2 và 2×1 vào hình trên sao cho không có phần nào của các viên gạch thừa ra ngoài, cũng không có vùng diện tích nào của hình chữ nhật không được lát.

Input: Dòng đầu ghi T là số lượng test ($T \leq 100$). T dòng tiếp theo mỗi dòng ghi một số n .

Output: Ghi ra T dòng là kết quả các test tương ứng. Vì con số này rất lớn nên các bạn chỉ cần in ra phần dư khi chia cho 10^9+7

Example:

input	output
3 1 2 3	1 2 3

153. Tính tổng

Program SUMK.*
Input SUMK.INP
Output SUMK.OUT
Score 100

Viết chương trình tính tổng:

$$1^k + 2^k + \dots + n^k$$

Input: Dòng đầu tiên ghi T ($T \leq 5$) - số lượng bộ dữ liệu. T dòng tiếp theo, mỗi dòng ghi hai số nguyên dương n, k ($k \leq 50, n \leq 10^9$)

Output: Gồm T dòng, mỗi dòng ghi kết quả tìm được (lấy phần dư khi chia cho 10^9+7)

Example:

input	output
2 1 1 2 3	1 9

154. Nhân ma trận

Program MAMU.*
Input MAMU.INP
Output MAMU.OUT
Score 100

Phép nhân ma trận chỉ thực hiện được khi số cột của ma trận bên trái bằng số dòng của ma trận bên phải. Nếu ma trận A có kích thước $m \times n$ và ma trận B có kích thước $n \times p$ thì ma trận tích $C = A \times B$ có kích thước $m \times p$, phần tử ở hàng i , cột j xác định bởi công thức:

$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$$

Để tính tích của ma trận A kích thước $m \times n$ với ma trận B kích thước $n \times p$ mất $m \times n \times p$ phép toán. Chú ý rằng phép nhân ma trận có tính chất kết hợp $(A \times B) \times C = A \times (B \times C)$ do đó thứ tự

thực hiện sẽ ảnh hưởng đến số lượng phép toán. Trong bài toán này, bạn cần tối ưu số phép toán để tính tích của k ma trận như sau:

$$P = A_1 \times A_2 \times \dots \times A_k$$

trong đó ma trận A_1 có kích thước $s_1 \times s_2$, ma trận A_2 có kích thước $s_2 \times s_3, \dots$, ma trận A_k có kích thước $s_k \times s_{k+1}$

Yêu cầu: Cho s_1, s_2, \dots, s_{k+1} hãy tính số phép toán ít nhất để thực hiện phép nhân trên.

Input:

- Dòng thứ nhất ghi số nguyên dương T - số lượng bộ dữ liệu ($T \leq 20$)
- Tiếp theo là T nhóm dòng, mỗi nhóm dòng có cấu trúc:
 - Dòng đầu là số nguyên dương k
 - Dòng tiếp theo chứa k số nguyên dương s_1, s_2, \dots, s_{k+1} ($s_i \leq 1000$)

Output: Gồm T dòng, mỗi dòng là kết quả tương ứng với một bộ dữ liệu

Example:

input	output
2	6
2	200
1 2 3	
3	
10 10 10 1	

Subtasks:

- Subtask 1: $k \leq 10$
- Subtask 2: $k \leq 100$

155. Tính tổng

Program SUM2N.*
 Input SUM2N.INP
 Output SUM2N.OUT
 Score 100

Cho số nguyên dương n, K . Tính:

$$S = (1 \cdot 2^0 + 2 \cdot 2^1 + \dots + n \cdot 2^{n-1}) \bmod K$$

Input: Gồm nhiều bộ dữ liệu, mỗi bộ cho trên một dòng chứa hai số nguyên dương n, K

Output: Gồm nhiều dòng, mỗi dòng là kết quả tương ứng với bộ dữ liệu vào.

Example:

Input	Output
1 10	1
2 2	1

Subtasks:

- Subtask 1: $n, k \leq 10^9$
- Subtask 2: $n, k \leq 10^{18}$

156. Lập phương

Program MMC.*
 Input MMC.INP
 Output MMC.OUT
 Score 100

Phép nhân hai ma trận chỉ được thực hiện khi số cột của ma trận bên trái bằng số hàng của ma trận bên phải. Nếu ma trận A có kích thước $m \times n$ và ma trận B có kích thước $n \times p$ thì ma trận tích $C = A \times B$ có kích thước $m \times p$, phần tử đứng ở hàng i , cột j xác định bởi:

$$c_{i,j} = a_{i,1} \times b_{1,j} + a_{i,2} \times b_{2,j} + \dots + a_{i,n} \times b_{n,j}$$

Ví dụ:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}; A^2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}; A^3 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

Yêu cầu: Cho ma trận A kích thước $n \times n$ và ma trận B kích thước $n \times n$. Hãy kiểm tra xem A^3 có bằng B hay không?

Input: Dòng thứ nhất ghi số nguyên dương T ($T \leq 20$) là số lượng bộ dữ liệu. Tiếp theo là T nhóm dòng mỗi nhóm dòng ứng với một bộ dữ liệu có dạng:

- Dòng đầu chứa số nguyên n
- n dòng sau, mỗi dòng chứa n số nguyên mô tả ma trận A . Các số có giá trị tuyệt đối không vượt quá 1000.
- n dòng tiếp theo, mỗi dòng chứa n số nguyên mô tả ma trận B . Các số có giá trị tuyệt đối không vượt quá 10^{18}

Output: Gồm T dòng, mỗi dòng tương ứng là kết quả ứng với một bộ dữ liệu theo thứ tự xuất hiện trong input: Ghi thông báo 'YES' nếu $A^3 = B$ và ghi 'NO' trong trường hợp ngược lại.

Example:

Input	Output
2	NO
2	YES
0 1	
1 1	
1 2	
2 2	
2	
0 1	
1 1	
1 2	
2 3	

Subtasks:

- Subtask 1: $n \leq 10$
- Subtask 2: $n \leq 500$

157. Cạnh 1

Program EDGE1.*
Input EDGE1.INP
Output EDGE1.OUT
Score 100

Cho ma trận kích thước $n \times m$ các phần tử chỉ nhận giá trị 0 hoặc 1.

Hãy tìm ma trận vuông con lớn nhất có các cạnh chỉ chứa 1.

0	1	1	1	1
0	1	0	1	1
1	1	0	0	1
1	1	1	1	1

Hình vuông lớn
nhất có cạnh chỉ 1

Input:

- Dòng đầu tiên chứa 2 số nguyên n và m ($1 \leq n, m \leq 1500$),
- Mỗi dòng trong n dòng sau chứa xâu độ dài m có các ký tự thuộc tập $\{0, 1\}$ xác định một dòng của ma trận.

Output: Một số nguyên – kích thước hình vuông con tìm được.

Example:

Input	Output
4 5	4

01111
01011
11001
11111

158. Chơi chữ

Program WORDS.*
 Input WORDS.INP
 Output WORDS.OUT
 Score 100

Io và Ao chơi một trò chơi trên các từ. Họ chỉ nói các từ được tạo thành từ các nguyên âm với qui tắc chữ cái đầu tiên của từ được nói tiếp theo phải trùng với chữ cái cuối cùng của từ được nói trước đó. Không được sử dụng từ nào quá 2 lần và chỉ được dùng các từ có trong một từ điển cho trước. Điểm của trò chơi là tổng độ dài của các từ được nói ra.

Viết chương trình xác định điểm lớn nhất của trò chơi nếu như chỉ sử dụng các từ có trong từ điển cho trước.

Input:

- Dòng đầu tiên chứa số tự nhiên N , $1 \leq N \leq 16$ xác định số lượng các từ có trong từ điển.
- N dòng tiếp theo mỗi dòng chứa 1 từ có trong từ điển. Mỗi từ là một dãy chỉ gồm các ký tự 'A', 'E', 'I', 'O', 'U'. Độ dài của mỗi từ không vượt quá 100. Tất cả các từ là khác nhau.

Output: Chứa một số nguyên duy nhất là điểm lớn nhất nhận được của trò chơi,

Example:

WORDS.INP	WORDS.OUT	WORDS.INP	WORDS.OUT	WORDS.INP	WORDS.OUT
3	8	4	13	5	16
AEIOU		AEEEO		IOO	
UIU		OEOAEEE		IUUO	
EO		AO		AI	
		O		OIOOI	
				AOOI	

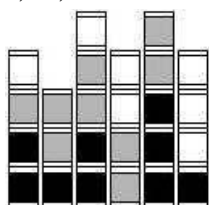
159. Bắn bia

Program SHOT.*
 Input SHOT.INP
 Output SHOT.OUT
 Score 100

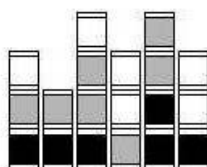
Trong ngày sinh nhật, Mirko nhận được khẩu súng bắn đạn hơi. Để thử khẩu súng này, cậu ta lấy các hộp chứa các hạt và đem đến sân chơi.

Có ba loại hộp đồ chơi: màu đen, trắng và xám, mỗi màu chứa một loại hạt khác nhau. Mirko xếp các hộp thành N cột, cột n đứng sau cột k ia. Trên mỗi cột thì các hộp màu đen xếp dưới cùng, tiếp theo là các hộp màu xám và màu trắng. Các cột có thể có độ cao khác nhau nhưng thứ tự màu các hộp phải đúng như mô tả trên, không nhất thiết trên một cột phải có đủ ba màu.

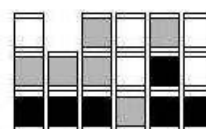
Sau khi xếp các hộp, Mirko thực hiện M phát đạn bắn theo phương ngang, mỗi viên đạn được bắn tại một độ cao nào đó. Các viên đạn này có sức công phá rất mạnh nên sau khi phá hủy hộp ở một độ cao nào đó của cột 1 nó tiếp tục phá hủy tất cả các hộp ở cùng độ cao trên tại các cột 2, 3, ..., N . Các hộp ở bên trên sẽ bị rơi xuống theo phương thẳng đứng. Các viên đạn được thực hiện theo thứ tự 1, 2, ..., M :



Trạng thái đầu



Viên 1: Độ cao



Viên 2: Độ cao

Trong trò chơi này, Mirko tự thưởng cho mình 1 điểm nếu hạ được 1 hộp màu đen, 2 nếu hạ 1 hộp màu xám và 5 nếu hạ được 1 hộp màu trắng.

Viết chương trình xác định số điểm mà Mirko nhận được

Input:

- Dòng đầu tiên ghi số nguyên N ($1 \leq N \leq 300000$) là số lượng cột
- Dòng thứ hai ghi N số mô tả độ cao các hộp màu đen trên mỗi cột
- Dòng thứ ba ghi N số mô tả độ cao các hộp xám trên mỗi cột.
- Dòng thứ tư ghi N số mô tả độ cao các hộp màu trắng trên mỗi cột

Các độ cao là số nguyên không âm có giá trị không vượt quá 10^6 .

- Dòng thứ năm ghi số nguyên M ($1 \leq M \leq 300000$) là số lượt bắn
- Dòng cuối cùng ghi M số nguyên lần lượt là độ cao của viên đạn trong mỗi lượt bắn. Độ cao là số nguyên không vượt quá 10^6 .

Output: Ghi ra M dòng, dòng thứ i là điểm mà Mirko nhận được ở lượt thứ i

Example:

Input	Output
6	12
2 1 2 0 3 1	7
1 2 2 2 2 0	
1 0 1 2 0 3	
2	
2 4	

160. Bao lồi dưới

Program LCONVEX.*
 Input LCONVEX.INP
 Output LCONVEX.OUT
 Score 100

Cho tập n đường thẳng $D = \{d_1, d_2, \dots, d_n\}$. Đường thẳng d_i có phương trình:

$$y = a_i \cdot x + b_i \quad (a_i \geq 0)$$

Bạn được cho trước m giá trị x_1, x_2, \dots, x_m . Hãy tính m giá trị y_1, y_2, \dots, y_m trong đó:

$$y_j = \min_{1 \leq i \leq n} \{a_i \cdot x_j + b_i\}$$

Input:

- Dòng đầu tiên ghi số nguyên dương $n \leq 10^5$
- n dòng tiếp theo, mỗi dòng chứa hai số thực a_i, b_i mô tả đường thẳng thứ i .
- Dòng tiếp theo chứa số nguyên $m \leq 10^5$
- m dòng cuối cùng, dòng thứ j chứa số thực x_j

Output: In ra m dòng, dòng thứ j chứa số thực y_j với 3 chữ số phần thập phân.

Example:

Input	Output
4	1.000
0.17 2.67	3.000
1 3	2.000
0.5 2	
0.8 4	
3	
-2	
2	
0	

161. Chặt cây

Program CUTTREE.*
 Input CUTTREE.INP

Output CUTTREE.OUT
Score 100

Có n cây $1, 2, \dots, n$ trong đó cây thứ i có chiều dài a_i . Cần phải chặt toàn bộ số cây này để mở rộng đường. Mỗi lần chặt một cây phải chặt hoàn toàn cây đó trước khi chuyển sang chặt cây khác. Để chặt một cây có chiều cao a_i , cưa máy cần có tối thiểu lượng pin là a_i . Máy sạc pin luôn đặt ở vị trí của cây có chỉ số lớn nhất đã bị chặt trước đó. Nếu vị trí đặt máy sạc là i thì chi phí sạc một đơn vị năng lượng là b_i . Khởi đầu cưa máy đã được sạc một lượng pin là 1. Tìm chi phí sạc pin tối thiểu để chặt toàn bộ n cây.

Input:

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 10^5$)
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($1 = a_1 \leq a_2 \leq \dots \leq a_n \leq 10^9$)
- Dòng thứ ba ghi n số nguyên b_1, b_2, \dots, b_n ($10^9 \geq b_1 \geq b_2 \geq \dots \geq b_n = 0$)

Output: Một số nguyên duy nhất là chi phí nhỏ nhất để chặt n cây

Example:

Input	Output
6 1 2 3 10 20 30 6 5 4 3 2 0	138

Giải thích: Một phương án chặt tối ưu là chặt các cây theo chỉ số lần lượt:

$$1 \rightarrow 3 \rightarrow 6 \rightarrow 2 \rightarrow 4 \rightarrow 5$$

- Ban đầu chặt cây 1 không mất chi phí (đủ pin)
- Để chặt cây 3, khi đó máy sạc pin ở vị trí 1, chi phí là $3 \times 6 = 18$
- Để chặt cây 6, khi đó máy sạc pin ở vị trí 3, chi phí là $30 \times 4 = 120$
- Để chặt cây 2, khi đó máy sạc pin ở vị trí 6, chi phí là $2 \times 0 = 0$
- Để chặt cây 4, khi đó máy sạc pin ở vị trí 6, chi phí là $3 \times 0 = 0$
- Để chặt cây 5, khi đó máy sạc pin ở vị trí 6, chi phí là $20 \times 0 = 0$

Tổng chi phí là $18 + 120 + 0 + 0 + 0 = 138$

162. Thuê đất

Program RENTING.*
Input RENTING.INP
Output RENTING.OUT
Score 100

Bờm vừa lập công lớn cho Phú Ông khi phát hiện ra kho báu được chôn trên đất nhà Phú Ông. Do vậy, Phú Ông đã cho Bờm thuê đất với giá đặc biệt. Vì biết Bờm là một người không thông minh nên Phú Ông đã nghĩ ra cách tính giá đất quái dị hòng lừa Bờm. Giá thuê đất được Phú Ông tính như sau: Giả sử Phú Ông có N ($1 \leq N \leq 50000$) miếng đất cho Bờm thuê với kích thước $l_i \times w_i$ với l_i là chiều dài và w_i là chiều rộng của các miếng đất ($1 \leq l_i \leq 10^6$; $1 \leq w_i \leq 10^6$). Bờm có thể nhóm các miếng đất thành một nhóm để thuê trong cùng một lượt. Số tiền thuê trong mỗi nhóm mà Bờm phải trả được tính bằng chiều dài lớn nhất của miếng đất trong nhóm nhân với chiều rộng lớn nhất của miếng đất trong nhóm. Chẳng hạn, nếu Bờm xếp hai miếng đất có kích thước (3,4) và (4,3) vào một nhóm thì số tiền phải trả cho nhóm này là $4 \times 4 = 16$ (xu).

Bờm muốn bạn giúp xem cần phải trả ít nhất là bao nhiêu để có thể thuê được toàn bộ các miếng đất của Phú Ông?

Input:

- Dòng đầu ghi số N
- Mỗi dòng trong N dòng sau ghi hai số nguyên w_i và l_i

Output: Một số nguyên duy nhất là số tiền nhỏ nhất mà Bờm phải trả cho Phú Ông.

Example:

Input	Output	Giải thích
4	500	Các nhóm
100 1		(100,1)
15 15		(15,15), (20,5)
20 5		(1,100)
1 100		Khi đó số tiền thuê là:
		$100 \cdot 1 + 15 \cdot 20 + 1 \cdot 100 = 500$

163. Điều chỉnh lương

Program	SALARY.*
Input	SALARY.INP
Output	SALARY.OUT
Score	100

Một công ty Startup tin học mới được thành lập và có $n + 2$ nhân viên được phân thành $n + 2$ cấp bậc từ thấp đến cao đánh số lần lượt $0, 1, 2, \dots, n, n + 1$. Nhân viên cấp 0 là nhân viên thường và có mức lương thấp nhất, nhân viên cấp $n + 1$ là CEO và có mức lương cao nhất; các nhân viên ở các cấp $1, 2, \dots, n$ sẽ có mức lương lần lượt là s_1, s_2, \dots, s_n . Một điều ngạc nhiên là với $i > j$ thì không nhất thiết $s_i > s_j$.

Sau một năm làm việc, tiểu ban khen thưởng của công ty quyết định thực hiện p_i lần tăng cấp cho nhân viên i ($i = 1 \div n$), tiểu ban kỷ luật quyết định thực hiện q_i lần giảm cấp cho nhân viên i .

Việc tăng cấp một lần cho nhân viên i được thực hiện bằng cách tăng lên đến cấp $k > i$ nhỏ nhất mà $s_k > s_i$; nếu nhân viên i được tăng cấp p_i lần thì quá trình trên lặp lại p_i lần, tuy nhiên chỉ được tăng lên tối đa là cấp $n + 1$.

Việc giảm cấp một lần cho nhân viên i được thực hiện bằng cách giảm xuống cấp $k < i$ lớn nhất mà $s_k < s_i$; nếu nhân viên i bị giảm cấp q_i lần thì quá trình trên thực hiện q_i lần, tuy nhiên chỉ được giảm đến cấp tối thiểu là 0 .

Có hai khả năng là tiểu ban khen thưởng làm việc trước rồi mới đến tiểu ban kỷ luật hoặc ngược lại. Mỗi nhân viên hiện có cấp $1, 2, \dots, n$ đều mong muốn lương của mình sang năm mới là lớn nhất có thể sau khi xử lý. Hỏi rằng với nhân viên hiện ở cấp i thì trong trường hợp lý tưởng đó, anh (cô) ta có cấp bậc là bao nhiêu?

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 5 \times 10^5$)
- Dòng thứ hai chứa n số nguyên s_1, s_2, \dots, s_n ($0 \leq s_i \leq 10^9; i = 1 \div n$)
- Dòng thứ ba chứa n số nguyên p_1, p_2, \dots, p_n ($0 < p_i \leq 10^9; i = 1 \div n$)
- Dòng thứ tư chứa n số nguyên q_1, q_2, \dots, q_n ($0 < q_i \leq 10^9; i = 1 \div n$)

Các số liên tiếp trong input cách nhau bởi dấu trống

Output: Một dòng chứa n số nguyên, số thứ i xác định bậc mới của nhân viên có bậc cũ là i ($i = 1 \div n$). Các số ghi cách nhau bằng dấu trống

Example:

Input	Output
10	8 11 4 3 11 4 11 1 11 11
6 2 4 8 5 1 4 9 8 7	
3 4 2 1 4 2 3 1 2 2	
2 1 4 2 3 3 1 3 1 1	

Giải thích: Ví dụ với nhân viên 8 (lương hiện tại $s_8 = 9$). Nếu thực hiện khen thưởng trước (1 lần) thì nhân viên này sẽ tăng đến bậc 11, sau khi kỷ luật (3 lần) nhân viên này bị giảm xuống cấp 6 ($11 \rightarrow 10 \rightarrow 7 \rightarrow 6$) có mức lương là 1; Tuy nhiên nếu thực hiện kỷ luật trước (3 lần), anh (cô) ta sẽ bị giảm về bậc 0 ($8 \rightarrow 7 \rightarrow 6 \rightarrow 0$) (lương thấp nhất), sau đó thực hiện khen thưởng (1 lần) anh (cô) ta

tăng lên bậc 1 có mức lương là 6. Tất nhiên anh (cô) ta muốn điều này hơn. Như vậy cấp bậc anh ta mong muốn là cấp 1

Subtasks:

- Subtask 1: $n \leq 10000$ [50%]
- Subtask 2: $n \leq 10^5$ [50%]

164. Đường cao tốc (HROADS)

Program	HROADS.*
Input	HROADS.INP
Output	HROADS.OUT
Score	100

Một đất nước có n thành phố với một hệ thống giao thông đảm bảo sự đi lại giữa hai thành phố bất kỳ. Để nâng cấp hệ thống giao thông quốc gia, Chính phủ quyết định nâng cấp một số con đường thành đường cao tốc. Để tiết kiệm kinh phí, việc nâng cấp chỉ thực hiện trên $n - 1$ con đường sao cho những con đường này đảm bảo sự đi lại giữa các thành phố và khoảng cách ngắn nhất từ thủ đô đến tất cả các thành phố theo các con đường cao tốc chính là khoảng cách ngắn nhất từ thủ đô đến các thành phố bằng hệ thống giao thông trước đây.

Yêu cầu: Hãy chọn ra $n - 1$ con đường để nâng cấp thành cao tốc thỏa mãn tiêu chí trên và tổng độ dài của tất cả các con đường là nhỏ nhất.

Input:

- Dòng đầu tiên chứa hai số nguyên dương n, m ($1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 3 \cdot 10^5$) là số thành phố và số con đường giao thông. Các thành phố được đánh số từ 1 đến n .
- m dòng tiếp theo, dòng thứ i chứa ba số nguyên u_i, v_i, w_i thể hiện một đường giao thông hai chiều nối thành phố u_i với thành phố v_i có độ dài w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9$). Dữ liệu đảm bảo rằng hệ thống giao thông đảm bảo sự đi lại giữa các thành phố và có không quá một đường giao thông nối trực tiếp giữa hai thành phố.
- Dòng cuối cùng chứa số nguyên u ($1 \leq u \leq n$) - số hiệu của thủ đô

Output:

- Dòng đầu tiên ghi một số nguyên là tổng độ dài của các con đường giữ lại
- Dòng tiếp theo in ra chỉ số các đường giao thông được giữ lại. Chỉ số của đường giao thông được đánh số từ 1 trở đi theo thứ tự xuất hiện trong input. Nếu như có nhiều phương án tối ưu thì chỉ cần in ra một trong số chúng.

Example:

Input	Output
3 3	2
1 2 1	1 2
2 3 1	
1 3 2	
3	

Ghi chú: Nếu chỉ đúng dòng đầu tiên thì được 50% số điểm của test

165. Đa giác

Program	CPOLY.*
Input	CPOLY.INP
Output	CPOLY.OUT
Score	100

Son muốn sử dụng các tấm gỗ có sẵn để ghép thành hàng rào xung quanh vườn. Hàng rào có dạng hình đa giác lồi (diện tích khác 0) được ghép từ các cạnh là các tấm gỗ đặt nằm ngang. Son không muốn cưa hay phá hỏng các tấm gỗ để làm hàng rào. Rõ ràng không phải từ các tấm gỗ bất kì nào cũng có thể ghép thành hàng rào đa giác lồi. Ví dụ từ 3 tấm gỗ có độ dài 10, 11, 12 có thể ghép thành

hàng rào tam giác, tuy nhiên từ 4 tấm gỗ có độ dài 100, 1, 2, 3 thì không thể ghép thành hàng rào tứ giác vì tấm gỗ dài 100 lớn hơn tổng độ dài của 3 tấm gỗ còn lại.

Son có n tấm gỗ. Son tự hỏi có bao nhiêu cách chọn ra các tấm gỗ từ các tấm đã có để từ chúng có thể ghép thành hàng rào đa giác lồi. **Hai cách chọn được gọi là khác nhau nếu như có một tấm gỗ thuộc cách chọn này nhưng không thuộc cách chọn kia.**

Yêu cầu: Bạn hãy giúp Son trả lời câu hỏi này. Do số lượng cách chọn có thể rất lớn nên bạn cần đưa ra số dư của số cách chọn trong phép chia cho $10^9 + 7$.

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 5000$) là số lượng tấm gỗ.
- Dòng thứ hai chứa n số nguyên l_i ($1 \leq l_i \leq 5000$) là độ dài các tấm gỗ.

Output: một dòng duy nhất là kết quả của bài toán.

Example:

Input	Output
5 1 2 3 4 6	7
4 5 5 5 5	5
5 10 1 2 3 50	0
6 12 16 14 8 17 7	40

Subtasks:

- Subtask 1: $n \leq 40$ [30%]
- Subtask 2: $n \leq 500$ [30%]
- Subtask 3: $n \leq 5000$ [40%]

166. Chặt cây (CUTTERS)

Program	CUTTERS.*
Input	CUTTERS.INP
Output	CUTTERS.OUT
Score	100

Có n cây dọc theo một con đường xem như trục số ở các tọa độ x_1, x_2, \dots, x_n . Cây thứ i có chiều cao là h_i . Khi chặt cây thứ i nó có thể đổ sang trái hoặc sang phải. Lúc đó thân cây sẽ chiếm một đoạn trục số là $[x_i - h_i, x_i]$ hoặc $[x_i, x_i + h_i]$. Tất nhiên nếu không chặt thì thân cây chỉ chiếm một điểm duy nhất (x_i) trên trục số. Chỉ có thể chặt một cây nếu như đoạn nó chiếm sau khi chặt không có bất kỳ một cây nào khác (đứng hay đã bị đổ).

Yêu cầu: Hãy tính số lượng lớn nhất các cây có thể chặt

Input:

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 10^5$)
- n dòng tiếp theo, dòng thứ i ghi hai số nguyên x_i, h_i ($1 \leq x_i, h_i \leq 10^9$) lần lượt là vị trí và độ cao của cây thứ i . Các cặp số được cho theo giá trị x_i tăng dần và đảm bảo không có hai cây nào ở cùng một vị trí

Output: Một số nguyên duy nhất là số lượng cây lớn nhất có thể chặt

Example:

Input	Output
5 1 2 2 1 5 10 10 9 19 1	3

167. Cặp đôi hoàn hảo

Program	PERFECT.*
Input	PERFECT.INP
Output	PERFECT.OUT
Score	100

XYZ là một công ty lớn trong lĩnh vực công nghệ phần mềm và tới đây họ chuẩn bị thực hiện một dự án lớn có thể thu lại lợi nhuận khổng lồ cho công ty. XYZ gồm n nhân viên, các nhân viên được đánh số từ 1 đến n , nhân viên thứ i có một chỉ số năng lực đúng bằng i . Tổ chức nhân sự của công ty XYZ có dạng đồ thị cây. Mỗi nhân viên có đúng một **cấp trên trực tiếp**, có một nhân viên duy nhất là tổng giám đốc, không ai là cấp trên của nhân viên này. Ta gọi nhân viên i là **cấp trên** của nhân viên j nếu nhân viên i là cấp trên trực tiếp của nhân viên j , hoặc nhân viên i là cấp trên của nhân viên u và nhân viên u là cấp trên trực tiếp của nhân viên j .

Để chuẩn bị tốt việc phân công công việc trong dự án lớn sắp tới, ban lãnh đạo công ty muốn đếm số cặp nhân viên hoàn hảo trong công ty. Hai nhân viên i và j là một cặp đôi hoàn hảo nếu họ thỏa mãn hai điều kiện sau:

- i là cấp trên của j .
- Chênh lệch năng lực giữa hai nhân viên không vượt quá k , tức là $|i - j| \leq k$.

Yêu cầu: Bạn hãy giúp ban lãnh đạo công ty XYZ đếm số cặp đôi hoàn hảo này.

Input:

- Dòng đầu tiên chứa 2 số nguyên n, k ($1 \leq n, k \leq 10^5$).
- $n - 1$ dòng tiếp theo, mỗi dòng chứa 2 số nguyên u, v ($1 \leq u, v \leq n$) miêu tả mối quan hệ nhân viên u là cấp trên trực tiếp của nhân viên v . Dữ liệu đảm bảo các mối quan hệ trong công ty tạo thành một cấu trúc cây.

Output: một dòng duy nhất là kết quả của bài toán.

Example:

Input	Output
5 2 3 2 3 1 1 4 1 5	4
10 2 1 4 1 5 2 8 2 9 3 1 3 2 6 3 7 6 8 10	12

Subtasks:

- Subtask 1 (30% điểm): $n \leq 300$.
- Subtask 2 (30% điểm): $n \leq 5000$.
- Subtask 3 (40% điểm): $n \leq 10^5$.

168. Số gần hoàn hảo

Program	QUASI.*
Input	QUASI.INP
Output	QUASI.OUT
Score	100

Trong bài toán này chúng ta quan tâm đến tổng các ước chẵn của một số tự nhiên. Các ước số chẵn của một số tự nhiên N là các số K nhỏ hơn hẳn N , sao cho N chia hết cho K . Ví dụ, tổng các ước số của 18 là:

$$S(18) = 1 + 2 + 3 + 6 + 9 = 21$$

Một số hoàn hảo $N > 0$ là số mà tổng các ước chẵn $S(N)$ của nó bằng chính N . Ví dụ số 6 và số 28 là các số hoàn hảo:

$$S(6) = 1 + 2 + 3 = 6$$

$$S(28) = 1 + 2 + 4 + 7 + 14 = 28$$

Thực chất các số hoàn hảo rất hiếm. Vì vậy chúng ta quan tâm đến các số gần hoàn hảo, nghĩa là tổng các ước chẵn của N không quá xa N .

Yêu cầu: Cho trước hai số tự nhiên L và D ($2 \leq L \leq 10^6$; $0 \leq D \leq 10^6$), hãy tìm số các số gần hoàn hảo nguyên dương nhỏ hơn L sao cho khoảng cách giữa số gần hoàn hảo và tổng các ước chẵn của nó không vượt quá D .

Ví dụ với $L = 10, D = 1$ ta tìm được 5 số gần hoàn hảo thỏa mãn yêu cầu: 1, 2, 4, 6 và 8.

Input:

- Dòng 1 ghi số nguyên dương $T \leq 10^5$ là số lượng bộ dữ liệu
- T dòng tiếp theo, dòng thứ i ghi hai số nguyên dương L_i, D_i là các giá trị L, D tương ứng với bộ dữ liệu thứ i

Output: Ghi ra T dòng, mỗi dòng ghi kết quả tìm được ứng một bộ dữ liệu trong input

Example:

Input	Output
1 10 1	4

Subtasks:

- Subtask 1: $T = 1$ [50%]
- Subtask 2: $T = 10^5$ [50%]

169. Số phần tử khác nhau

Program	DIFFERENT.*
Input	DIFFERENT.INP
Output	DIFFERENT.OUT
Score	100

Cho dãy n số nguyên a_1, a_2, \dots, a_n và m truy vấn, mỗi truy vấn có dạng (u, v) thể hiện yêu cầu bạn phải đếm số phần tử khác nhau của dãy con a_u, a_{u+1}, \dots, a_v

Input:

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 3 \times 10^5$)
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9; i = 1 \div n$)
- Dòng thứ ba ghi số nguyên m ($1 \leq m \leq 3 \times 10^5$)
- m dòng cuối, mỗi dòng ghi hai số nguyên u, v ($1 \leq u \leq v \leq n$) thể hiện một truy vấn

Output: Với mỗi truy vấn u, v in ra số phần tử khác nhau của dãy con a_u, a_{u+1}, \dots, a_v trên một dòng.

Example:

Input	Output
5 1 1 2 1 3 3 1 5 2 4 3 5	3 2 3

Subtasks:

- Subtask 1: $n, m \leq 1000$ [50%]
- Subtask 2: $n, m \leq 3 \times 10^5$ [50%]

170. Số lượng lớn hơn

Program XGREAT.*
 Input XGREAT.INP
 Output XGREAT.OUT
 Score 100

Cho dãy n số nguyên a_1, a_2, \dots, a_n và m truy vấn, mỗi truy vấn có dạng một bộ ba số nguyên (i, j, x) thể hiện yêu cầu bạn phải đếm số lượng phần tử lớn hơn x trong dãy con a_i, a_{i+1}, \dots, a_j

Input:

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 3 \times 10^5$)
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$; $i = 1 \div n$)
- Dòng thứ ba ghi số nguyên dương m ($1 \leq m \leq 3 \times 10^5$)
- m dòng cuối, mỗi dòng ghi bộ ba số nguyên i, j, x ($1 \leq i \leq j \leq n$; $1 \leq x \leq 10^9$) thể hiện một truy vấn

Output: Với mỗi truy vấn in kết quả tìm được trên một dòng

Example:

Input	Output
5	2
5 1 2 3 4	0
3	3
2 4 1	
4 4 4	
1 5 2	

171. Ma trận

Program MATRIX.*
 Input MATRIX.INP
 Output MATRIX.OUT
 Score 100

Lực lượng kháng chiến do Neo lãnh đạo chống lại đội quân rô bốt đã trưởng thành, đủ sức chủ động mở những cuộc tiến công mới. Bản đồ vùng rô bốt kiểm soát có dạng ma trận $\{0, 1\}$ gồm n dòng và m cột. Theo sự đánh giá của Ban lãnh đạo kháng chiến vùng có thể tấn công phải là một ma trận con (tức là phần của ma trận bị chặn bởi hai đường ngang và 2 đường dọc) thỏa mãn các điều kiện:

- ✚ Có ít nhất 2 hàng,
- ✚ Dòng thứ nhất của vùng này phải trùng với dòng cuối cùng, dòng thứ 2 – trùng với dòng trước dòng cuối, ... dòng thứ i từ trên xuống phải trùng với dòng thứ i từ dưới lên.

Hình bên nêu 2 trong số 18 vùng có thể tấn công.

Hãy xác định số lượng vùng có thể tấn công.

01001	01001
00110	00110
01011	01011
10011	10011

Input:

- ✚ Dòng đầu tiên chứa 2 số nguyên n và m ($2 \leq n, m \leq 400$),
- ✚ Mỗi dòng trong n dòng sau chứa xâu độ dài m chỉ chứa các ký tự trong tập $\{0, 1\}$ mô tả một dòng của bản đồ.

Output: Một số nguyên – số lượng vùng có thể tấn công.

Example:

input	output
4 5 01001 00110	18

01011	
10011	

Subtasks:

- Subtask 1: $n, m \leq 30$ [10 tests]
- Subtask 2: $n, m \leq 400$ [10 tests]

172. Biến đổi số

Program NUMTRANS.*
 Input NUMTRANS.INP
 Output NUMTRANS.OUT
 Score 100

Cho số nguyên dương D và hai quy tắc biến đổi số sau:

- Số X biến đổi thành $k \times X$ với $1 \leq k \leq D$
- Số X biến đổi thành X/k với $1 \leq k \leq D$ và X chia hết cho k .

Hãy xác định số bước biến đổi ít nhất cần thực hiện để biến đổi từ số A thu được số B .

Input: Ba số nguyên A, B, D ($1 \leq A, B, D \leq 10^9$)

Output: Số nguyên kết quả

Example:

input	output
3 5 5	2
1 8 2	3

173. Dãy xâu (SEQSTR)

Program SEQSTR.*
 Input SEQSTR.INP
 Output SEQSTR.OUT
 Score 100

Cho một dãy gồm n xâu s_1, s_2, \dots, s_n và một số nguyên dương k . Một cặp hai xâu s_i và s_j trong dãy được gọi là tương thích với nhau nếu thỏa mãn:

- $0 < j - i \leq k$,
- Hai xâu s_i và s_j có cùng độ dài.

Yêu cầu: Hãy xác định số cặp các xâu tương thích với nhau trong dãy các xâu đã cho.

Input:

- Dòng đầu chứa hai số nguyên n và k ($3 \leq n \leq 300.000$; $1 \leq k \leq n$).
- n dòng tiếp theo mỗi dòng chứa một xâu có độ dài từ 2 đến 20 kí tự gồm các chữ cái tiếng Anh in hoa.

Output: một dòng duy nhất là kết quả của bài toán.

Example:

Input	Output
4 2 OTN ABC THA HUN	5
6 3 CFETHIA LLOYD STEVIE KEVIN MALCABC DABNEY	2

Subtasks:

- Subtask 1 (40%): $n \leq 5000$.

- Subtask 2 (60%): $n \leq 300.000$.

174. Trò chơi với số

Program	NGAME.*
Input	NGAME.INP
Output	NGAME.OUT
Score	100

Có hai sỹ quan chơi một trò chơi. Đầu tiên người thứ nhất đưa cho người thứ hai một số nguyên dương n . Nhiệm vụ của người thứ hai là thực hiện nhiều vòng chơi nhất có thể. Mỗi vòng chơi, anh ta chọn ra một số nguyên $x > 1$ sao cho n chia hết cho x và thay thế n bởi n/x . Khi n bằng 1 thì trò chơi kết thúc và điểm mà người chơi nhận được là số lần lấy được số x .

Để trò chơi trở nên thú vị hơn, số n được chọn có dạng $\frac{a!}{b!}$ với hai số nguyên dương a, b ($a \geq b$). Ở đây $k! = 1 \times 2 \times \dots \times k$

Hãy xác định số điểm lớn nhất mà người thứ hai nhận được.

Input:

- Dòng đầu tiên ghi số nguyên dương t ($1 \leq t \leq 10^6$) là số lần chơi
- t dòng tiếp theo, mỗi dòng ghi hai số nguyên dương a, b ($1 \leq b \leq a \leq 5 \cdot 10^6$) mô tả dạng của số n trong ván chơi.

Output: Với mỗi lần chơi in ra số điểm lớn nhất nhận được

Example:

Input	Output
2	2
3 1	5
6 3	

Subtasks:

- Subtask 1: $1 \leq b \leq a \leq 14$; $T \leq 10$ [50%]
- Subtask 2: $20 \leq b \leq a \leq 5 \cdot 10^6$; $T \leq 10^6$ [50%]

175. Màu của cây

Program	COLTREE.*
Input	COLTREE.INP
Output	COLTREE.OUT
Score	100

Bạn được cho trước một cây (đồ thị liên thông không chu trình) gồm N nút đã được tô màu. Mỗi màu được đại diện bởi một số nguyên thuộc khoảng $[1, 10^9]$. Hãy tìm số lượng màu phân biệt trong cây con gốc S ?

Input:

- Dòng đầu chứa ba số nguyên N ($1 \leq N \leq 10^5$) – số nút trên cây, M ($1 \leq M \leq 10^5$) – số lượng truy vấn và R ($1 \leq R \leq N$) – gốc của cây.
 - Trong $N - 1$ dòng tiếp theo, mỗi dòng chứa 2 số nguyên phân biệt miêu tả một cạnh trên cây.
 - N dòng tiếp theo, dòng thứ $N + i$ chứa một số nguyên thuộc khoảng $[1, 10^9]$ là màu của nút thứ i .
 - M dòng tiếp theo, dòng thứ $2N + i$ chứa một số nguyên là đỉnh tương ứng với truy vấn thứ i .
- Các số trong dữ liệu đầu vào được viết cách nhau ít nhất một khoảng trắng.

Output: Ghi đúng M dòng, dòng thứ i chứa một số nguyên là kết quả tương ứng với truy vấn thứ i .

Example:

Input	Output
4 2 1	3
1 2	2

2	4
2	3
10	
20	
20	
30	
1	
2	

Chú ý: 20% số điểm có $1 \leq N, M \leq 1000$.

176. Đổi chỗ hoán vị

Program SWAP.*
 Input SWAP.INP
 Output SWAP.OUT
 Score 100

Bạn được cho một hoán vị p của các số $1, 2, \dots, n$ và cặp m vị trí (a_j, b_j) .

Tại mỗi bước bạn có thể chọn một cặp trong m cặp và đổi chỗ các số ở các vị trí đó. Hãy tìm hoán vị có thứ tự từ điển lớn nhất mà bạn có thể nhận được.

Cho p và q là hai hoán vị của các số $1, 2, \dots, n$. Ta nói q có thứ tự từ điển lớn hơn p nếu tồn tại vị trí i ($1 \leq i \leq n$) sao cho $q_k = p_k$ với $1 \leq k < i$ và $q_i > p_i$.

Input: Dòng đầu tiên chứa hai số nguyên n và m ($1 \leq n, m \leq 10^5$) là số phần tử của hoán vị p và số cặp vị trí. Dòng thứ hai chứa n số nguyên phân biệt p_i ($1 \leq p_i \leq n$) là các phần tử của hoán vị p . Mỗi dòng trong m dòng tiếp theo chứa hai số nguyên a_j, b_j ($1 \leq a_j, b_j \leq n$) mô tả một cặp vị trí đổi chỗ. Chú ý rằng đây là các vị trí, không phải là các giá trị của các phần tử đổi chỗ.

Output: Ghi ra một dòng n số nguyên phân biệt q_i ($1 \leq q_i \leq n$) là hoán vị có thứ tự từ điển lớn nhất tìm được.

Example:

Input	Output
3 2 1 3 2 1 3 3 1	2 3 1
9 6 1 2 3 4 5 6 7 8 9 1 4 4 7 2 5 5 8 3 6 6 9	7 8 9 4 5 6 1 2 3

Subtasks:

- Subtask 1 (30%): $1 \leq n, m \leq 10$.
- Subtask 2 (30%): $1 \leq n, m \leq 10^3$.
- Subtask 3 (40%): $1 \leq n, m \leq 10^5$.

177. Dãy ngoặc đúng

Program BRACKET.*
 Input BRACKET.INP
 Output BRACKET.OUT
 Score 100

Xét dãy ngoặc chỉ gồm các ký tự “(” và “)”. Chúng ta gọi một dãy ngoặc là dãy ngoặc đúng nếu tồn tại một cách chèn “+” và “1” vào dãy ngoặc để được một biểu thức toán học hợp lệ. Ví dụ, các dãy ngoặc “(())()”, “()” và “((()())” là các dãy ngoặc đúng, còn dãy ngoặc “(“(“, “(())” và “(())(” không là dãy ngoặc đúng.

Cho một chuỗi S gồm các ký tự “(” và “)”. Bạn hãy tìm chuỗi con dài nhất của chuỗi S mà nó là dãy ngoặc đúng và đếm xem có bao nhiêu chuỗi con như vậy.

Input: Gồm một dòng chứa chuỗi S ($1 \leq |S| \leq 10^6$, ở đó $|S|$ là độ dài chuỗi S), chỉ bao gồm các ký tự “(” và “)”.

Output: Ghi ra trên một dòng hai số nguyên ngăn cách nhau một dấu cách tương ứng là độ dài của chuỗi con dài nhất mà nó là dãy ngoặc đúng và số chuỗi con như vậy. Nếu không có chuỗi con nào như vậy thì ghi ra một dòng chứa “0 1”.

Example:

Input	Output
) ((())) (() ()	6 2
)) (0 1

Subtasks:

- Subtask 1 (30%): $1 \leq |S| \leq 300$.
- Subtask 2 (30%): $1 \leq |S| \leq 5000$.
- Subtask 3 (40%): $1 \leq |S| \leq 10^6$.

178. Cắt cây

Program	CUT.*
Input	CUT.INP
Output	CUT.OUT
Score	100

Cây là một đồ thị vô hướng liên thông và không có chu trình. Bạn được cho một cây với N nút được đánh số từ 1 đến N và một số K . Nút i có một trọng số $W[i]$.

Hãy chia cây thành tối đa K cây con bằng cách loại bỏ tối đa $K-1$ cạnh sao cho trọng số của cây con lớn nhất là nhỏ nhất. Trọng số của cây con là tổng trọng số của tất cả các nút trong cây con đó. Bạn chỉ cần tìm giá trị trọng số nhỏ nhất của cây con có trọng số lớn nhất mà có thể đạt được bằng cách loại bỏ tối đa $K-1$ cạnh.

Input: Dòng đầu tiên chứa 2 số nguyên N ($1 \leq N \leq 20.000$) và K ($1 \leq K \leq N$) tương ứng là số nút của cây và số cây con. Dòng thứ hai chứa n số nguyên $W[i]$ ($1 \leq W[i] \leq 10^9$) là trọng số của các nút. Tiếp theo có $N-1$ dòng, mỗi dòng chứa 2 số nguyên mô tả hai đầu mút của một cạnh.

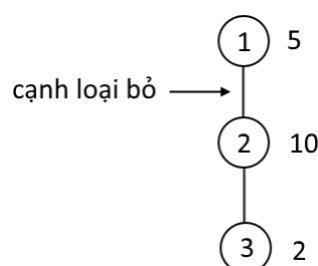
Output: Ghi ra một số nguyên là kết quả của bài toán.

Example:

Input	Output
3 2 5 10 2 1 2 2 3	12
3 3 5 10 2 1 2 1 3	10

Ví dụ 1: Chúng ta loại bỏ cạnh giữa nút 1 và nút 2, thu được 2 cây con:

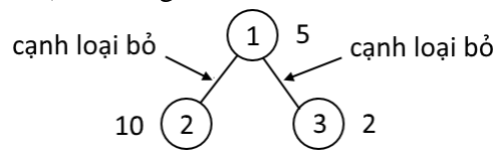
- Cây con thứ nhất (gồm nút 1) có trọng số là 5.
- Cây con thứ hai (gồm 2 nút: nút 2 và nút 3) có trọng số là $10 + 2 = 12$.



Cây con có trọng số lớn nhất là 12. Đây là cách chia cho cây con có trọng số lớn nhất là nhỏ nhất.

Ví dụ 2: Chúng ta loại bỏ cả 2 cạnh và thu được 3 cây con:

- Cây con thứ nhất (gồm nút 1) có trọng số là 5.
- Cây con thứ hai (gồm nút 2) có trọng số là 10.
- Cây con thứ ba (gồm nút 3) có trọng số là 2.



Cây con có trọng số lớn nhất là 10. Đây là cách chia cho cây con có trọng số lớn nhất là nhỏ nhất.

Subtasks:

- Subtask 1 (30%): $1 \leq N \leq 25$.
- Subtask 2 (30%): $1 \leq N \leq 5000$.
- Subtask 3 (40%): $1 \leq N \leq 20.000$.

179. Tương đồng

Program	SIMILA.*
Input	SIMILA.INP
Output	SIMILA.OUT
Score	100

Cho hai dãy số nguyên $A = (a_1, a_2, \dots, a_n)$ và $B = (b_1, b_2, \dots, b_m)$. Mức độ giống nhau của hai dãy số được cho bởi số nguyên k lớn nhất sao cho tồn tại hai số nguyên dương p, q sao cho các dãy con $(a_p, a_{p+1}, \dots, a_{p+k-1})$ và $(b_q, b_{q+1}, \dots, b_{q+k-1})$ là hoán vị của nhau ($1 \leq p \leq n - k + 1, 1 \leq q \leq m - k + 1$).

Hãy xác định mức độ giống nhau của hai dãy đã cho.

Input:

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 1000$)
- Dòng thứ hai ghi n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^5; i = 1 \div n$)
- Dòng thứ ba ghi số nguyên dương m ($1 \leq m \leq 1000$)
- Dòng thứ tư chứa m số nguyên dương b_1, b_2, \dots, b_m ($1 \leq b_j \leq 10^5, j = 1 \div m$)

Output: Ghi ba số nguyên k, p, q trong đó k là mức độ giống nhau của hai dãy số còn p, q là hai chỉ số sao cho $(a_p, a_{p+1}, \dots, a_{p+k-1})$ và $(b_q, b_{q+1}, \dots, b_{q+k-1})$ là hoán vị của nhau. Nếu mức độ tương đồng là 0 thì ghi ba số nguyên 0, -1, -1

Example:

Input	Output
3	3 1 1
1 2 3	2 1 3
3	
2 1 3	

Subtasks:

- Subtask 1: $n, m \leq 300$
- Subtask 2: $n, m \leq 1000$

180. Du lịch Tràng An

Program	CRUISE.*
Input	CRUISE.INP
Output	CRUISE.OUT
Score	100

Tràng An (Ninh Bình) là một khu du lịch nổi tiếng trong cả nước gồm nhiều hang động được nổi với

nhau bằng các con sông nhỏ. Du khách sẽ đi dọc theo các con sông này để tham quan các hang động. Để đảm bảo an toàn giao thông, các thuyền du lịch chỉ được đi theo một chiều qui định trước.

Cụ thể, hệ thống sông trong khu vực hang động có N ngã ba sông (đánh số từ 1 đến N). Tại mỗi ngã ba sông, người ta xây dựng các cảng, mỗi cảng có hai chỉ dẫn tiếp theo, một chỉ dẫn hướng đến cảng bên trái (L) và một chỉ dẫn hướng đến cảng bên phải. Tour du lịch cho khách luôn xuất phát từ cảng 1, đi theo M chỉ dẫn chỉ gồm các ký tự 'L' (bên trái) và 'R' (bên phải).

Trong dịp khuyến mại cho các cặp tình nhân, ban tổ chức quyết định mỗi cặp tình nhân được đi K lần theo chỉ dẫn trên (với chi phí bằng 1 lần). Hằng Nga tự hỏi, nếu nhận tour khuyến mại này thì bên cuối cuộc hành trình là cảng nào?

Viết chương trình trả lời câu hỏi trên của Chị Hằng.

Input:

- Dòng đầu tiên ghi ba số nguyên dương N, M và K ($N \leq 1000, M \leq 500, K \leq 10^9$)
- N dòng tiếp theo, mỗi dòng ghi hai số nguyên là số hiệu của cảng tiếp theo nếu xuất phát từ cảng i đi theo hướng trái và hướng phải.
- Dòng cuối cùng chứa M ký tự cách nhau bởi dấu trống chỉ gồm hai ký tự 'L' và 'R' là các chỉ dẫn của tour du lịch

Output: Một số nguyên duy nhất là số hiệu của cảng cuối cùng.

Example:

input	output
4 3 3 2 4 3 1 4 2 1 3 L L R	4

Subtasks:

- Subtask 1: $K \leq 100$
- Subtask 2: $K \leq 100000$
- Subtask 3: $K \leq 10^9$

181. Xuất hiện hai lần

Program	DTWICE.*
Input	DTWICE.INP
Output	DTWICE.OUT
Score	100

Cho dãy n số nguyên a_1, a_2, \dots, a_n và m truy vấn. Mỗi truy vấn có dạng hai số nguyên u, v với ý nghĩa đếm xem trong dãy con a_u, a_{u+1}, \dots, a_v có bao nhiêu giá trị khác nhau mà các giá trị này xuất hiện đúng hai lần?

Input:

- Dòng đầu ghi hai số nguyên dương n, m ($1 \leq n, m \leq 5 \times 10^5$)
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9 : i = 1 \div n$)
- m dòng cuối, mỗi dòng ghi hai số nguyên u, v mô tả một truy vấn.

Output: Với mỗi truy vấn in ra một dòng một số nguyên - kết quả tìm được.

Example:

Input	Output
5 1 1 2 1 1 1 1 3	1

Ghi chú: Có đến 40% số điểm của bài ứng với các test có $n, m \leq 5000$

182. Tổng các chữ số

Program	SUMDIGIT.*
Input	SUMDIGIT.INP
Output	SUMDIGIT.OUT
Score	100

Bozo là một cậu bé kỳ lạ. Hàng ngày cậu ta luôn trăn trở với những câu hỏi kỳ lạ. Câu hỏi hôm nay là: có bao nhiêu số nguyên trong đoạn $[A, B]$ có tổng chữ số của nó bằng S và số nhỏ nhất là số nào? Viết chương trình trả lời hộ Bozo câu hỏi này để cậu bé còn đi ngủ.

Input: Gồm một dòng duy nhất chứa ba số nguyên A, B, S ($1 \leq A \leq B \leq 10^{15}$; $1 \leq S \leq 135$). Dữ liệu vào luôn đảm bảo rằng giá trị nhỏ nhất của dòng đầu là 1.

Output:

- Dòng đầu tiên chứa số lượng số nguyên trong đoạn $[A, B]$ có tổng các chữ số của nó bằng S
- Dòng thứ hai chứa số nguyên nhỏ nhất trong số đó

Example:

input	output
1 100 10	9
	19

Ghi chú: Mỗi dòng trả lời đúng được 50% số điểm của test (kết quả nhất thiết phải có hai dòng)

183. Số nhị phân

Program	BNUM.*
Input	BNUM.INP
Output	BNUM.OUT
Score	100

Cho hai số nguyên không âm A, B ($0 \leq A \leq B \leq 10^{18}$). Hãy xác định xem trong đoạn $[A, B]$ có bao nhiêu số mà trong dạng biểu diễn nhị phân của nó có đúng K ($0 \leq K \leq 18$) chữ số 0 có nghĩa?

Ví dụ, $A = 1, B = 18, K = 3$ có 3 số:

- 8 = 1000
- 17 = 10001
- 18 = 10010

Input: Một dòng chứa ba số nguyên A, B, K

Output: Ghi số lượng số tìm được

Example:

Input	Output
1 18 3	3

184. Số 68

Program	NUM68.*
Input	NUM68.INP
Output	NUM68.OUT
Score	100

Một số nguyên dương N ($N > 1$) luôn có thể biểu diễn được dưới dạng tổng của hai số nguyên dương A và B ($N = A + B, A \leq B$). Trong bài toán này chúng ta sẽ quan tâm đến cách biểu diễn một số nguyên N thành tổng hai số nguyên A, B thỏa mãn tính chất: Trong biểu diễn của A hoặc của B phải chứa chữ số 6 hoặc chữ số 8.

Ví dụ, $N = 10$, có tất cả 5 cách biểu diễn nhưng chỉ có hai cách thỏa mãn là $2+8$; $4+6$.

Yêu cầu: Cho số nguyên dương N ($N > 1$). Hãy đếm số cách biểu diễn N thành tổng hai số nguyên dương A và B thỏa mãn tính chất: Trong biểu diễn của A hoặc B phải chứa chữ số 6 hoặc chữ số 8.

Input: Gồm nhiều dòng, mỗi dòng ứng với một số nguyên N ($1 < N \leq 10^{18}$)

Output: Gồm nhiều dòng, mỗi dòng là kết quả tương ứng với dữ liệu vào.

Example:

input	output
10	2
19	4

185. Số 13

Program N13.*
 Input N13.INP
 Output N13.OUT
 Score 100

John không hề thích con số 13 vì anh ta cho rằng đó là số không may mắn. Trong một lần phải liệt kê các số tự nhiên từ A đến B , John muốn lọc ra các số mà trong dạng biểu diễn của nó không xuất hiện số 13. Ví dụ số 111639786 không xuất hiện số 13, còn số 113 thì có xuất hiện số 13.

Yêu cầu: Cho A, B . Hãy xác định số lượng các số trong đoạn $[A, B]$ mà trong dạng biểu diễn của nó không xuất hiện số 13.

Input: Gồm nhiều dòng, mỗi dòng chứa hai số nguyên A, B ($0 \leq A \leq B \leq 10^{15}$)

Output: Gồm nhiều dòng, mỗi dòng là số lượng số tìm được ứng với file dữ liệu vào.

Example:

input	output
1 13	12
100 1000	882

186. Số đối xứng

Program PALINN.*
 Input PALINN.INP
 Output PALINN.OUT
 Score 100

Xét các số tự nhiên có N chữ số thỏa mãn tính chất sau:

- Là số đối xứng
- Số đó chia hết cho M

Ví dụ với $N = 3, M = 9$ ta có 10 số 171, 252, 333, 414, 585, 666, 747, 828, 909, 999 là số đối xứng chia hết cho 9.

Yêu cầu: Cho N, M, K

- Hãy đếm số đối xứng có N chữ số chia hết cho M
- Nếu sắp xếp các số theo giá trị tăng dần thì số thứ K là số nào?

Input:

- Dòng thứ nhất ghi hai số N, M ($2 < N < 40, 0 < M < 100$)
- Dòng thứ hai gồm một số K

Output:

- Dòng đầu là số lượng số được xét
- Dòng thứ hai gồm một số K

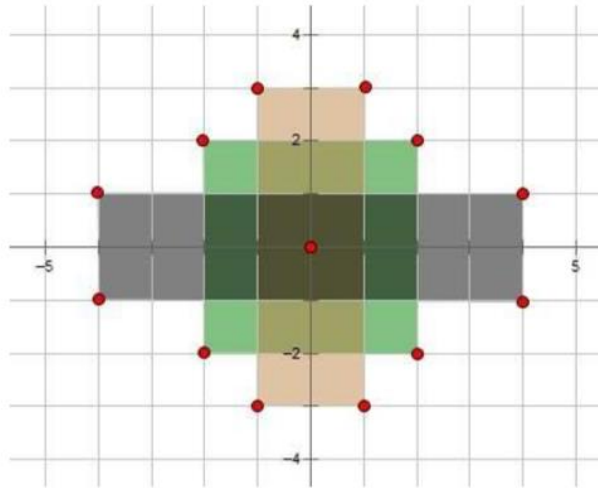
Example:

input	output
3 9	10
3	333

187. Phân bị phủ

Program COVER.*
 Input COVER.INP
 Output COVER.OUT
 Score 100

Trên mặt phẳng tọa độ cho n hình chữ nhật có tâm (giao điểm của hai đường chéo) là gốc tọa độ. Các cạnh song song với một trong hai trục tọa độ và có chiều dài là số nguyên chẵn.



Tính diện tích của phần mặt phẳng bị phủ bởi các hình chữ nhật trên?

Input:

- Dòng đầu tiên ghi số nguyên dương n ($1 \leq n \leq 10^6$)
- n dòng tiếp theo, mỗi dòng ghi hai số nguyên dương chẵn X, Y lần lượt là chiều dài của cạnh song song với trục Ox và chiều dài của cạnh song song với trục Oy ($2 \leq X, Y \leq 10^7$)

Output: Một số nguyên duy nhất là phần diện tích của mặt phẳng bị phủ bởi các hình chữ nhật

Example:

Input	Output
3 8 2 4 4 2 6	28

Subtasks:

- Subtask 1: Tất cả các số trong input không quá 4000 [40%]
- Subtask 2: Không có hình chữ nhật nào nằm hoàn toàn bên trong hình khác [50%]

188. Chia kẹo (SWEET3)

Program	SWEET3.*
Input	SWEET3.INP
Output	SWEET3.OUT
Score	100

Ba anh em An, Bình, Cường có n chiếc kẹo và muốn chia n chiếc kẹo thành 3 phần, sau đó mỗi người sẽ nhận một phần. An rất yêu thích số a và vì đó là ngày sinh của An, do đó An sẽ chỉ nhận phần có số kẹo chia hết cho a , thậm chí An không nhận kẹo nếu không có phần nào có số kẹo là bội của a . Tương tự Bình và Cường có số yêu thích tương ứng là b và c và cũng chỉ nhận phần kẹo có số kẹo là bội của số mình thích. Vì vậy ba anh em quyết định tìm cách tác số n thành các số nguyên không âm x, y, z thỏa mãn yêu cầu sau:

1. $x + y + z = n$
2. x chia hết cho a ; y chia hết cho b ; z chia hết cho c

Yêu cầu: Cho 4 số nguyên dương n, a, b, c . Hãy đếm số cách tách n thành 3 số nguyên không âm x, y, z thỏa mãn yêu cầu của ba anh em.

Dữ liệu: Vào từ file văn bản SWEETS3.INP

- Dòng đầu tiên chứa số nguyên dương T là số bộ dữ liệu.
- T dòng tiếp theo, mỗi dòng chứa 4 số nguyên dương n, a, b, c ($n \leq 10^9$; $a, b, c \leq 31$)

Kết quả: Ghi ra file văn bản SWEETS3.OUT gồm T dòng, mỗi dòng là kết quả tương ứng với bộ dữ liệu vào.

Ví dụ:

Input	Output
2	3
10 3 3 4	0
10 6 6 6	

Subtasks:

- Subtask 1: $T \leq 3$ [50%]
- Subtasks 2: $T \leq 1000$ [50%]

189. Kẻ hủy diệt

Program	TERMINATOR.*
Input	TERMINATOR.INP
Output	TERMINATOR.OUT
Score	100

"Kẻ hủy diệt" là seri phim nổi tiếng của điện ảnh Mỹ, Trong bộ phim này Skynet-máy tính siêu đẳng đã thống trị Thế giới và quân kháng chiến, dưới dự lãnh đạo của John Connor quyết tâm tiêu diệt chúng. Theo kế hoạch này họ sẽ tấn công và chiếm đóng một số thành phố hiện đang nằm trong quyền kiểm soát của Skynet.

Có n thành phố đánh số từ 1 đến n và m con đường hai chiều nối giữa các thành phố này. Hiện tại Skynet đang chiếm đóng tất cả các thành phố. Có k thành phố là căn cứ của Skynet và không thể bị chiếm trong mọi trường hợp. Vì vậy, quân kháng chiến chỉ có thể chiếm một tập hợp không rỗng các thành phố không phải là căn cứ.

Sau khi chiếm đóng, quân kháng chiến phải bảo vệ các thành phố chiếm được khỏi các cuộc tấn công. Nếu một thành phố được nối trực tiếp với một vài thành phố do Skynet kiểm soát thì nó rất dễ bị phản công. Do vậy Connor định nghĩa độ an toàn của một thành phố là tỷ lệ giữa số thành phố lân cận nằm trong quyền kiểm soát của quân kháng chiến với tổng số thành phố lân cận. Ở đây hai thành phố lân cận được hiểu là có đường nối trực tiếp với nhau.

Độ an toàn của một tập không rỗng S các thành phố mà quân kháng chiến có thể chiếm được định nghĩa là độ an toàn của thành phố kém an toàn nhất.

Yêu cầu: Tìm một tập S khác rỗng các thành phố mà quân kháng chiến có thể chiếm sao cho độ an toàn của tập này là lớn nhất

Input:

- Dòng đầu tiên ghi ba số nguyên n, m, k ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$, $1 \leq k \leq n - 1$)
- Dòng thứ hai ghi k số nguyên phân biệt mô tả các thành phố là căn cứ của Skynet
- Tiếp theo là m dòng mô tả các con đường. Dòng thứ i chứa hai số nguyên a_i, b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$). Tất cả các thành phố đều có ít nhất một thành phố lân cận.

Output:

- Dòng đầu tiên ghi số nguyên r - số lượng thành phố của tập S tìm được.
- Dòng thứ hai ghi r số nguyên là các thành phố của S , có thể in theo một trình tự bất kỳ. Chú ý rằng không được có bất kỳ thành phố nào là căn cứ của Skynet.

Nếu có nhiều phương án, bạn chỉ cần in ra một phương án bất kỳ.

Example:

input	output
9 8 4	3
3 9 6 8	1 4 5
1 2	
1 3	
1 4	
1 5	

2 6	
2 7	
2 8	
2 9	

Subtasks

- Subtask 1: $n \leq 20$ [30%]
- Subtask 2: $n \leq 5000$ [30%]
- Subtask 2: $n \leq 10^5$ [40%]

190. Phục vụ nhà hàng

Program	PIZZA.*
Input	PIZZA.INP
Output	PIZZA.OUT
Score	100

Phú Ông thành lập cửa hàng chuyên bán gạo, Bờm và Cuội thành các nhân viên vận chuyển gạo đến tận nhà cho những người mua. Hôm nay, cả hai phải chuyển gạo đến nhà của n người mua trong làng đánh số từ 1 đến n . Tùy vào mức độ thân thiện nên đối với người mua thứ i , nếu Bờm giao gạo anh ta sẽ được một lượng tiền hoa hồng là a_i còn nếu Cuội giao thì số tiền này là b_i . Tất nhiên với mỗi địa chỉ thì chỉ có một người đi giao. Ngoài ra, vì lý do sức khỏe nên Bờm chỉ có thể giao không quá X người mua và Cuội chỉ có thể giao không quá Y người mua. May mắn thay $X + Y \geq n$.
Hãy tìm cách phân chia nhiệm vụ giao gạo đến n người sao cho tổng số tiền hoa hồng mà Bờm và Cuội nhận được là lớn nhất?

Input:

- Dòng đầu tiên chứa ba số nguyên n, X, Y ($1 \leq n \leq 10^5; 1 \leq X, Y \leq n; X + Y \geq n$)
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$)
- Dòng thứ ba chứa n số nguyên b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^4$)

Output: Một số nguyên duy nhất là tổng tiền hoa hồng lớn nhất mà cả hai nhận được

Example:

input	output
5 3 3 1 2 3 4 5 5 4 3 2 1	21

Subtasks:

- Subtask 1: $n \leq 20$ [30%]
- Subtask 2: $n \leq 5000$ [40%]
- Subtask 3: $n \leq 10^5$ [30%]

191. Tái cơ cấu (RESTRUCT)

Program	RESTRUCT.*
Input	RESTRUCT.INP
Output	RESTRUCT.OUT
Score	100

Ngay cả những công ty thành công nhất cũng có thể trải qua một giai đoạn khủng hoảng khi phải thực hiện một quyết định khó khăn - tái cơ cấu, loại bỏ, hợp nhất các phòng ban, và những công việc khó chịu khác. Hãy xem xét mô hình sau của một công ty:

Con n người làm việc trong một công ty phần mềm lớn. Mỗi người đều thuộc một phòng nào đó. Đầu tiên, mỗi người có một dự án riêng và ở trong một phòng riêng (khởi đầu công ty có n phòng, mỗi phòng có đúng một người).

Tuy nhiên, thời điểm khắc nghiệt đã đến với công ty và công ty phải thuê một người quản lý khủng hoảng, nhiệm vụ của người này là xây dựng lại quá trình làm việc để tăng tính hiệu quả. Chúng ta sử

dụng khái niệm một đội bao gồm nhiều người cùng làm việc. Người quản lý khủng hoảng có thể đưa ra các quyết định thuộc một trong hai loại:

1. Nhập phòng của đội chứa nhân viên x với phòng của đội chứa nhân viên y thành một phòng lớn hơn với số nhân viên là số nhân viên của hai phòng nhập lại. Nếu như đội chứa nhân viên x và đội chứa nhân viên y là một thì không có điều gì xảy ra.
2. Nhập tất cả các đội chứa nhân viên x , đội chứa nhân viên $x + 1, \dots$, đội chứa nhân viên y thành một đội lớn hơn. Ở đây x và y ($1 \leq x \leq y \leq n$) là số hiệu của hai nhân viên nào đó.

Đôi khi người quản lý khủng hoảng tự hỏi xem hai nhân viên x và y ($1 \leq x, y \leq n$) có cùng một đội hay không?

Viết chương trình giúp người quản lý khủng hoảng thực hiện các điều trên.

Input:

- Dòng đầu tiên chứa hai số nguyên n, q ($1 \leq n \leq 2 \cdot 10^5; 1 \leq q \leq 5 \cdot 10^5$) là số nhân viên và số thao tác mà người quản lý khủng hoảng thực hiện.
- q dòng tiếp theo, mỗi dòng mô tả một thao tác gồm ba số $type, x, y$ với $type \in \{1, 2, 3\}$. Nếu $type = 1$ hoặc $type = 2$ thì đây là thao tác đưa ra quyết định thuộc một trong hai loại. Nếu $type = 3$ thì đây là thao tác xác định xem x, y có thuộc cùng một đội hay không? (chú ý rằng trong mọi trường hợp có thể xảy ra $x = y$)

Output: Với các thao tác $type = 3$ in ra 'YES' hoặc 'NO' (không có dấu nháy) tùy theo hai người tương ứng có cùng một đội hay không?

Example:

Input	Output
8 6	NO
3 2 5	YES
1 2 5	YES
3 2 5	
2 4 7	
2 1 2	
3 1 7	

Subtasks:

- Subtask 1: $n, q \leq 1000$ [30%]
- Subtask 2: Chỉ có quyết định loại 1; $n, q \leq 2 \cdot 10^5$ [20%]
- Subtask 3: Chỉ có quyết định loại 2; $n, q \leq 2 \cdot 10^5$ [20%]
- Subtask 3: $n, q \leq 2 \cdot 10^5$ [30%]

192. Số cách đi

Program	SHELFDRV.*
Input	SHELFDRV.INP
Output	SHELFDRV.OUT
Score	100

Để thử nghiệm một xe tự hành, người ta tiến hành một thí nghiệm cho xe vượt qua các chướng ngại vật và đi đến đích. Có thể mô tả thí nghiệm này như sau:

Có một lưới ô vuông kích thước $m \times n$, các hàng đánh số từ 1 đến m từ trên xuống dưới và các cột đánh số từ 1 đến n từ trái qua phải; ô nằm ở giao của hàng i với cột j ký hiệu là (i, j) . Trên một số ô có vật cản không thể đi qua được, các ô còn lại không có vật cản. Xe tự hành xuất phát từ ô $(1, 1)$. Mỗi bước nó chỉ có thể di chuyển đến ô chung cạnh ở bên phải hoặc bên dưới nếu như các vị trí này còn nằm trong lưới và không có vật cản. Chính xác hơn từ ô (i, j) xe chỉ có thể di chuyển đến ô $(i, j + 1)$ hoặc $(i + 1, j)$ nếu như các ô này còn trong lưới và không có vật cản. Xe tự hành cần di chuyển đến ô (m, n) .

Viết chương trình đếm xem có bao nhiêu cách khác nhau để di chuyển xe tự hành từ ô (1,1) đến ô (m,n) theo các qui tắc trên? Hai cách đi được coi là khác nhau nếu như có ít nhất một ô có trên hành trình của cách đi này nhưng không có trên hành trình của cách đi kia.

Input:

- Dòng đầu tiên ghi ba số nguyên dương m, n, k với m, n là số hàng và số cột của lưới; k là số lượng ô có vật cản ($1 \leq m, n \leq 10^5; 0 \leq k \leq 2000$)
- k dòng tiếp theo, dòng thứ i ghi hai số nguyên (r_i, c_i) thể hiện (r_i, c_i) là ô chứa vật cản ($1 \leq r_i \leq m; 1 \leq c_i \leq n$)

Dữ liệu đảm bảo rằng các ô (1,1) và (m,n) là các ô không có vật cản.

Output: một số nguyên duy nhất là số lượng cách di chuyển khác nhau của xe tự hành từ ô (1,1) đến ô (m,n). Vì con số này có thể rất lớn nên chỉ cần in ra phần dư của nó khi chia cho 10^9+7 .

Ví dụ:

Input	Output
3 4 2 2 2 2 3	2

Ghi chú:

- Có 40% số test ứng với 40% số điểm có $1 \leq m, n, k \leq 1000$
- Có 20% số test ứng với 20% số điểm có $k = 0; 10^4 \leq m, n \leq 10^5$
- Có 10% số test ứng với 10% số điểm có $k = 1; 10^4 \leq m, n \leq 10^5$
- Có 30% số test ứng với 30% số điểm có $1 \leq k \leq 2000; 10^4 \leq m, n \leq 10^5$

193. Biểu thức ngoặc 1

Program	BTN1.*
Input	BTN1.INP
Output	BTN1.OUT
Score	100

Biểu thức ngoặc là xâu chỉ gồm các ký tự '(' và ')'. Biểu thức ngoặc đúng và bậc của biểu thức ngoặc được định nghĩa một cách đệ qui như sau:

- Biểu thức rỗng là biểu thức ngoặc đúng và có bậc bằng 0
- Nếu A là biểu thức ngoặc đúng có bậc k thì (A) cũng là biểu thức ngoặc đúng có bậc $k + 1$
- Nếu A và B là hai biểu thức ngoặc đúng có bậc k_1, k_2 thì AB cũng là biểu thức ngoặc đúng có bậc $\max(k_1, k_2)$

Ví dụ '()()' là biểu thức ngoặc đúng có bậc bằng 2 còn '(()())' là biểu thức ngoặc đúng có bậc 3

Yêu cầu: Cho S là một xâu chỉ gồm các ký tự '(', ')' và '?'. Hãy tìm cách thay thế các ký tự '?' bằng các ký tự '(' hoặc ')' để nhận được xâu T là biểu thức ngoặc đúng có bậc lớn nhất.

Input: Gồm một dòng chứa xâu ký tự S (độ dài xâu không vượt quá 666) chỉ gồm các ký tự '(', ')', '?'. Dữ liệu luôn đảm bảo có nghiệm

Output: Ghi ra một số nguyên là bậc lớn nhất của biểu thức ngoặc tìm được.

Example:

input	output
(?? (?) ?	3
????	2

Ghi chú: Có 50% test ứng với 50% điểm của bài có độ dài xâu S không vượt quá 50

194. Biểu thức ngoặc 2

Program	BTN2.*
Input	BTN2.INP
Output	BTN2.OUT
Score	100

Biểu thức ngoặc là chuỗi chỉ gồm các ký tự '(' và ')'. Biểu thức ngoặc đúng và bậc của biểu thức ngoặc được định nghĩa một cách đệ quy như sau:

- Biểu thức rỗng là biểu thức ngoặc đúng và có bậc bằng 0
- Nếu A là biểu thức ngoặc đúng có bậc k thì (A) cũng là biểu thức ngoặc đúng có bậc $k + 1$
- Nếu A và B là hai biểu thức ngoặc đúng có bậc k_1, k_2 thì AB cũng là biểu thức ngoặc đúng có bậc $\max(k_1, k_2)$

Ví dụ '()()' là biểu thức ngoặc đúng có bậc bằng 2 còn '(()())' là biểu thức ngoặc đúng có bậc 3

Yêu cầu: Cho S là một chuỗi chỉ gồm các ký tự '(', ')' và '?'. Hãy tìm cách thay thế các ký tự '?' bằng các ký tự '(' hoặc ')' để nhận được chuỗi T là biểu thức ngoặc đúng có bậc lớn nhất.

Input: Gồm một dòng chứa chuỗi ký tự S (độ dài chuỗi không vượt quá 666) chỉ gồm các ký tự '(', ')', '?'. Dữ liệu luôn đảm bảo có nghiệm

Output: Ghi ra chuỗi T tìm được, nếu như có nhiều chuỗi T thì in ra chuỗi có thứ tự từ điển lớn nhất.

Chú ý rằng '(' < ')'.

Example:

input	output
(? ? (?) ?	() ((()))
????	(())

Ghi chú: Có 50% test ứng với 50% điểm của bài có độ dài chuỗi S không vượt quá 50

195. Biểu thức ngoặc 3

Program	RESTORE.*
Input	RESTORE.INP
Output	RESTORE.OUT
Score	100

Biểu thức ngoặc là chuỗi chỉ gồm các ký tự '(' và ')'. Biểu thức ngoặc đúng là biểu thức ngoặc được định nghĩa một cách đệ quy như sau:

- Biểu thức rỗng là biểu thức ngoặc đúng có bậc bằng 0
- Nếu A là biểu thức ngoặc đúng có bậc bằng k thì (A) cũng là biểu thức ngoặc đúng có bậc bằng $k + 1$
- Nếu A và B là hai biểu thức ngoặc đúng có bậc tương ứng là k_1 và k_2 thì AB cũng là biểu thức ngoặc đúng có bậc là $\max(k_1, k_2)$

Ví dụ, '()()' là biểu thức ngoặc đúng có bậc bằng 2 còn '(()())' là biểu thức ngoặc đúng có bậc bằng 3

Yêu cầu: Cho số nguyên k và chuỗi S chỉ gồm các ký tự '(', ')' và '?'. Hãy đếm số cách thay các ký tự ? bằng các ký tự '(' hoặc ')' để được biểu thức ngoặc đúng có bậc đúng bằng k .

Input:

- Dòng đầu ghi số nguyên dương k
- Dòng thứ hai ghi chuỗi S (độ dài chuỗi không vượt quá 200) ghi gồm các ký tự '(', ')' và '?'

Output: Gồm một dòng là số cách thay ký tự '?' trong chuỗi S thành ký tự '(' hoặc ')' để nhận được chuỗi T là biểu thức ngoặc đúng có độ sâu k

Example:

input	output
2 ???? (?)	1

196. Biểu thức ngoặc 4

Program	BTN.*
Input	BTN.INP
Output	BTN.OUT
Score	100

Biểu thức ngoặc là chuỗi chỉ gồm các ký tự '(', ')', '[', ']', '{', '}'. Biểu thức ngoặc đúng và bậc của biểu thức ngoặc được định nghĩa một cách đệ quy như sau:

- Biểu thức rỗng là biểu thức ngoặc đúng có bậc bằng 0
- Nếu A là biểu thức ngoặc đúng có bậc bằng k thì (A) , $[A]$, $\{A\}$ cũng là biểu thức ngoặc đúng có bậc $k + 1$
- Nếu A, B là hai biểu thức ngoặc đúng có bậc k_1, k_2 thì AB cũng là biểu thức ngoặc đúng có bậc $\max(k_1, k_2)$

Ví dụ, $'()()'$ là biểu thức ngoặc đúng có bậc bằng 2 còn $'\{()\{\}\}'$ là biểu thức ngoặc đúng có bậc bằng 3.

Yêu cầu: Cho n, k và S là một chuỗi chỉ gồm các ký tự '(', ')', '[', ']', '{', '}', '?'. Hãy đếm số cách thay thế ký tự '?' thành một trong các ký tự '(', ')', '[', ']', '{', '}' để nhận được chuỗi T là biểu thức ngoặc đúng có bậc đúng bằng k .

Input:

- Dòng đầu ghi hai số nguyên n, k
- Dòng thứ hai chứa một chuỗi có độ dài n

Output: Gồm một dòng chứa số nguyên là số cách thay thế ký tự '?' thành một trong các ký tự '(', ')', '[', ']', '{', '}' để được biểu thức ngoặc đúng có bậc đúng bằng k

Example:

input	output
2 1	1
[?]	

Subtasks:

- Subtask 1: $n \leq 20$
- Subtask 2: $n \leq 100$

197. Sửa lỗi (FIXES.*)

Program	FIXES.*
Input	FIXES.INP
Output	FIXES.OUT
Score	100

Bài tập về nhà của Mai là một chuỗi ký tự thể hiện một biểu thức ngoặc đúng và nhiệm vụ của Mai là phải thêm các số, các phép toán vào giữa các dấu '(' và ')' để được một biểu thức toán học đúng.

Tuy nhiên do vội vàng nên khi chép chuỗi ký tự vào vở, Mai đã chép nhầm một số ký tự: Ký tự '(' thì ghi thành ')' và ngược lại, ký tự ')' ghi thành '('.

Hãy giúp Mai sửa một số ít nhất các ký tự trong chuỗi ký tự của mình để được một biểu thức ngoặc đúng (có thể không phải là biểu thức ngoặc đúng mà cô giáo đã ra nhưng điều này không quan trọng!!!)

Có nhiều cách để định nghĩa một biểu thức ngoặc đúng. Cách đơn giản nhất có lẽ là: *Biểu thức ngoặc đúng là biểu thức mà số lượng ký tự '(' bằng số lượng ký tự ')' và với mỗi vị trí i trong chuỗi ký tự thì số lượng ký tự '(' không nhỏ hơn số lượng ký tự ')' tính từ vị trí 0 đến vị trí i .*

Input: Một chuỗi ký tự chỉ gồm '(' và ')' có độ dài không quá 100000 ký tự

Output: Một số nguyên duy nhất là số lượng ít nhất các ký tự cần phải đổi

Example:

Input	Output
()) (2

198. Biểu thức ngoặc đồng thời

Program	BEAULC.*
Input	BEAULC.INP

Output BEAULC.OUT
Score 100

Cho K dãy ký tự có cùng độ dài N bao gồm chỉ các ký tự '(' và ')'. Một cặp số (i, j) được gọi là một cặp vị trí đẹp nếu như chỉ xét các ký tự từ vị trí thứ i đến vị trí thứ j của K xâu ký tự đã cho thì các xâu ký tự này lập thành biểu thức ngoặc đúng.

Ví dụ, xét 3 xâu ký tự sau:

S₀ =) () ((())) (())
S₂ = () (() () ((())
S₃ =))) (() ()) (())
 1 2 3 4 5 6 7 8 9 0 1 2 3 4

Khi đó cặp (4,9) là một cặp vị trí đẹp vì 3 xâu con của 3 xâu ký tự ứng với cặp này:

((()))
() () ()
(() ())

là các biểu thức ngoặc đúng. Tương tự các cặp (11,14) và (12,13) cũng là các cặp vị trí đẹp

Yêu cầu: Đếm số lượng các cặp vị trí đẹp

Input:

- Dòng 1: Ghi hai số nguyên dương K, N ($K \leq 10$, $N \leq 50000$)
- K dòng tiếp theo, mỗi dòng ghi một xâu ký tự chỉ gồm '(' và ')'

Output: Một số nguyên duy nhất là số cặp vị trí đẹp tìm được

Example:

Input	Output
3 14) () ((())) (()) () (() () ((())))) (() ()) (())	3

199. Duyệt trại

Program BIVOUAC.*
Input BIVOUAC.INP
Output BIVOUAC.OUT
Score 100

Nhân dịp kỷ niệm 20 năm Tái lập Tỉnh Hải Dương, học sinh hai trường A và B tổ chức cắm trại trên một khu đất rộng. Có m trại của học sinh trường A được đánh số 1, 2, ..., m và n trại của học sinh trường B được đánh số 1, 2, ..., n . Để đơn giản có thể xem vị trí các trại như là các điểm trên mặt phẳng tọa độ hai chiều (không nhất thiết phải phân biệt!!!)

Đề chấm điểm, thầy Trưởng Ban xuất phát từ trại 1 của trường A và lần lượt thăm $m + n$ trại của cả hai trường và kết thúc ở trại m của trường A sao cho nếu theo danh sách này thì các trại của trường A được thăm theo trình tự 1, 2, ..., m và các trại của trường B được thăm theo trình tự 1, 2, ..., n .

Khi di bộ từ trại này sang trại khác với khoảng cách D thì thầy Trưởng ban tốn D^2 kilocalor. Hãy viết chương trình giúp thầy tính mức tiêu hao năng lượng tối thiểu nếu như thăm đủ $m + n$ trại của hai trường theo trình tự thỏa mãn yêu cầu nêu trên.

Input:

- Dòng 1: Chứa hai số nguyên dương $m, n \leq 1000$
- m dòng tiếp theo, dòng thứ i ghi tọa độ trại thứ i của trường A ($i = 1, 2, \dots, m$).
- n dòng tiếp theo, dòng thứ j ghi tọa độ trại thứ j của trường B ($j = 1, 2, \dots, n$)

Tất cả các tọa độ đều là các số nguyên có giá trị trong khoảng 0...1000

Output: một dòng ghi một giá trị là tổng năng lượng tiêu hao tối thiểu tìm được (dễ thấy con số này luôn là số nguyên !!!)

Example:

Input	Output
3 2 0 0 1 0 2 0 0 3 1 3	20

200. Chọn đội

Program TEAM.*
 Input TEAM.INP
 Output TEAM.OUT
 Score 100

Nhân dịp đón chào năm mới, nhà trường tổ chức cuộc thi đấu cờ vua giữa các lớp. Lớp 12 Tin đã lập ra một đội gồm n bạn và lớp 12 Toán cũng lập ra được một đội gồm m bạn. Qua khảo sát trước đó mỗi bạn trong cả hai đội có một hệ số elo (nguyên) được tính trước. Thể thức thi đấu như sau: Mỗi lớp sẽ chọn ra một đội hình gồm k bạn, sau đó bạn có hệ số elo cao nhất ở lớp 12 Tin sẽ đấu với bạn có hệ số elo cao nhất ở lớp 12 Toán, bạn có hệ số elo cao thứ nhì ở lớp 12 Tin sẽ đấu với bạn có hệ số elo cao thứ nhì của lớp 12 Toán,... Đội 12Tin sẽ được coi là thắng hiển nhiên nếu như trong tất cả các cặp đấu hệ số elo của bạn lớp 12 Tin cao hơn hệ số elo của bạn lớp 12 Toán.

Yêu cầu: Hãy đếm xem có bao nhiêu phương án lập đội hình khác nhau của lớp 12 Tin và 12 Toán sao cho đội 12 Tin thắng hiển nhiên. Ở đây mỗi tập hợp k bạn lớp 12 tin và k bạn lớp 12 toán được coi là một phương án lập đội hình. Con số này có thể lớn nên chỉ cần in phần dư của kết quả tìm được khi chia cho 10^9+9 .

Input:

- Dòng 1: Ghi ba số nguyên dương n, m, k ($1 \leq n, m \leq 1000; 1 \leq k \leq 10$). Số k không lớn hơn n hoặc m .
- Dòng 2: Ghi n số nguyên là hệ số elo của các bạn lớp 12 Tin
- Dòng 3: Ghi m số nguyên là hệ số elo của các bạn lớp 12 Toán

Output: kết quả tìm được

Example:

Input	Output
10 10 3 1 2 2 6 6 7 8 9 14 17 1 3 8 10 10 16 16 18 19 19	382