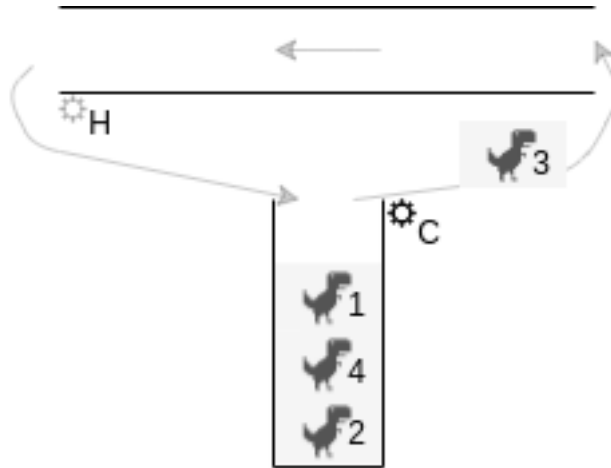


Bài A. SDINO

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Khủng long kỳ thực không hề hung dữ như người ta vẫn tưởng, chúng rất thông minh và dễ bảo. Bạn là người trông giữ khủng long và có nhiệm vụ sắp xếp lại khủng long trong chuồng. Chuồng chỉ có một cửa ra vào, và chiều ngang hẹp chỉ vừa đủ cho một con khủng long di chuyển, vì thế con nào vào chuồng trước thì sẽ phải ra sau. Ở cạnh chuồng có một hành lang. Hành lang có một cửa vào và một cửa ra, và chiều ngang cũng vừa đủ cho một con khủng long di chuyển, nên con nào vào trước sẽ phải ra trước.



Có n con khủng long, tất cả đều đang ở trong chuồng. Con khủng long thứ i (tính từ cửa chuồng vào trong) có số hiệu p_i (p_1, p_2, \dots, p_n là một hoán vị của $\{1, 2, \dots, n\}$). Để sắp xếp lại các con khủng long, bạn sẽ sử dụng đèn báo hiệu được lắp ở chuồng và hành lang. Khi bật đèn báo hiệu ở chuồng, nếu trong chuồng có khủng long thì một con khủng long trong chuồng sẽ đi sang hành lang. Khi bật đèn báo hiệu ở hành lang, nếu ở hành lang có khủng long thì một con khủng long ở hành lang sẽ đi vào chuồng.

Yêu cầu: Hãy chỉ ra một dãy không quá 10^6 báo hiệu để sắp xếp lại các con khủng long theo thứ tự từ nhỏ đến lớn tính từ cửa chuồng vào trong.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương n
- Dòng tiếp theo chứa n số nguyên dương $p_1 p_2 \dots p_n$

Kết quả

Ghi chuỗi s ($0 \leq |s| \leq 10^6$) gồm nhiều ký tự viết liền nhau, các ký tự C cho biết bạn bật đèn ở chuồng, các ký tự H cho biết bạn bật đèn ở hành lang. Lưu ý các đèn là đèn báo hiệu và sẽ tắt ngay sau khi được bật sáng. Các đèn được bật theo đúng trình tự trong chuỗi s

Ví dụ

stdin	stdout
4 3 1 4 2	CCHHCCCCCHHHHCCHHCCHCCHCCHHHH

Hạn chế

- Subtask 1: $n \leq 50$
- Subtask 2: $n \leq 1000$
- Subtask 3: $n \leq 20000$

Bài B. BROADGAME

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Tý và Tí là đôi bạn thân cùng đội tuyển. Một hôm họ rảnh rỗi đã nghĩ ra một trò chơi đối kháng như sau:

- Cho k bảng 2 chiều chứa các phân số, bảng thứ i có kích thước $n_i \times m_i$. Ô có tọa độ (i, j) sẽ ghi phân số $\frac{i}{j}$
- Hai người luân phiên thực hiện lượt chơi, Tý đi trước. Đến lượt mình, người chơi chọn ra một bảng trong số k bảng trên, chọn một ô trên bảng và tô màu nó, sao cho ô được chọn có giá trị lớn hơn mọi ô khác đã được tô màu trên bảng đó
- Đến lượt mình, ai không thực hiện được lượt chơi nữa thì thua. Cả hai người đều rất thông minh - có lẽ vì học đội tuyển chẳng

Hãy giúp Tý xác định rằng mình có thể thắng trong trò chơi này hay không

Dữ liệu vào

Gồm 3 bộ dữ liệu trong 1 test, mỗi bộ bao gồm:

- Dòng đầu: k
- k dòng tiếp theo, dòng thứ i ghi: $n_i \ m_i$

Kết quả

- Gồm 3 dòng ghi YES/NO tương ứng là Tý có/không thể thắng

Ví dụ

stdin	stdout
5	NO
1 1	YES
1 2	NO
2 3	
3 4	
4 5	
3	
7 7	
8 8	
9 9	
4	
2 2	
3 3	
4 4	
4 5	

Hạn chế

- Subtask 0: $1 \leq n, m, k \leq 200$
- Subtask 1: $1 \leq n, m \leq 2000, k \leq 20000$
- Subtask 2: $1 \leq n, m, k \leq 20000, n = m$
- Subtask 3: $1 \leq n, m, k \leq 20000$

Bài C. CANDIES

Ban tổ chức đã chuẩn bị rất nhiều kẹo cho các thí sinh, như là phần thưởng cho những nỗ lực không ngừng nghỉ của họ. Kẹo được ban tổ chức chia thành các gói để phân phát cho thí sinh. Số cái kẹo trong mỗi gói kẹo luôn luôn là số fibonacci, và số lượng gói kẹo mỗi loại mà ban tổ chức có là vô hạn. Số fibonacci được định nghĩa như sau:

$$\begin{aligned}f_1 &= 1 \\f_2 &= 1 \\f_k &= f_{k-1} + f_{k-2} \quad \forall k \geq 3\end{aligned}$$

Giả sử có n thí sinh tham gia cuộc thi. Sau cuộc thi, các thí sinh sẽ được xếp hạng từ 1 đến n (không có hai thí sinh nào cùng hạng). Thí sinh hạng thứ i sẽ nhận được một số gói kẹo sao cho tổng lượng kẹo trong các gói đúng bằng $n-i+1$. Ban tổ chức đã chọn cách phát kẹo sao cho đối với mỗi thí sinh số gói kẹo nhận được là ít nhất.

Có một vấn đề là việc chia kẹo rất tốn thời gian, vì thế những người nhận nhiều hơn hoặc bằng k gói kẹo sẽ được nhận kẹo của mình vào ngày hôm sau.

Yêu cầu: Hãy tính tổng số gói kẹo của những người được nhận kẹo vào ngày hôm sau.

Dữ liệu vào

- Dòng đầu chứa số testcase: T .
- T dòng tiếp theo mỗi dòng chứa hai số n, k .

Kết quả

Gồm T dòng là kết quả cho T testcase theo thứ tự đầu vào.

Ví dụ

test	answer
3	4
6 2	2
4 2	6
5 1	

Giải thích

Ở testcase 3, thí sinh hạng hai nhận 2 gói và bốn thí sinh còn lại mỗi người nhận 1 gói, tất cả đều nhận vào hôm sau.

Hạn chế

- $1 \leq T \leq 10^5$, $1 \leq n, k \leq 10^{15}$ trong tất cả các test
- 20% số test với $n, k \leq 10^5$, $T \leq 100$
- 20% số test với $n, k \leq 10^5$, $T > 100$
- 30% số test với $n > 10^5$, $T \leq 100$
- 30% số test còn lại không có ràng buộc gì thêm

Bài D. VIRUS

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Hùng đang phát triển một phần mềm diệt virus dựa trên nguyên tắc so khớp mẫu. Theo đó, cậu có tập S các xâu nhị phân là mẫu thường gặp trong mã nguồn của virus (mã độc). Mỗi tệp tin f có thể được hiểu như một xâu nhị phân, độ tương thích của f với mẫu x là số lần xuất hiện của x ở trong f (xuất hiện theo nghĩa bằng với một đoạn con liên tiếp). Để đánh giá f có bị nhiễm virus hay không, Hùng muốn tính tổng độ tương thích của f với mọi xâu trong S .

Trong quá trình sử dụng, Hùng có thể cập nhật thêm các mẫu virus mới. Ban đầu Hùng chưa có mẫu virus nào được báo cáo. Cụ thể bạn cần thực hiện hai loại truy vấn sau:

- Loại một: 0 x : Thêm xâu x vào tập S . Nếu x đã xuất hiện trong S , Hùng vẫn thêm x vào (theo cậu, càng có nhiều người dùng báo cáo một mẫu thì mẫu đó càng quan trọng). Có thể hiểu S là một multiset.
- Loại hai: 1 f : Tính toán và đưa ra tổng số lần xuất hiện của các xâu trong S trên xâu f .

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương Q
- Q dòng tiếp theo mỗi dòng mô tả một truy vấn: 0 x hoặc 1 f

Kết quả

- Với mỗi truy vấn loại hai, in ra kết quả trên một dòng

Ví dụ

stdin	stdout
5	10
0 0	11
0 1	
0 01	
1 10011001	
1 01010011	

Hạn chế

- $1 \leq Q \leq 10^6$, tổng độ dài tất cả các xâu đầu vào không quá 2×10^6
- Subtask 0: $|S| \leq 10$
- Subtask 1: Các truy vấn loại hai nằm liên tiếp nhau
- Subtask 2: Ràng buộc gốc

Bài E. CAMERA

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Đất nước HCNV có n thành phố được đánh số từ 1 đến n , một số cặp thành phố được nối với nhau bằng đường hai chiều. Bọn cướp ở đây thường thực hiện các phi vụ của mình ở một trong k con đường đặc biệt. Để tránh bị chốt chặn, k con đường này được chọn sao cho đôi một không có thành phố chung (có thể hiểu k đường này lập thành một cặp ghép). Sau khi tìm ra các tuyến đường đặc biệt, nhà nước quyết định cấp cho bộ giao thông đúng k camera với dự định lắp mỗi cái trên một tuyến đường đặc biệt nhằm giám sát tội phạm

Sau khi bàn bạc, bộ giao thông quyết định sẽ đưa ra một phương án tốt hơn để tối đa hóa lợi ích từ k camera này. Theo đó, họ sẽ lắp camera tại các thành phố (thay vì các tuyến đường). Khi thành phố x được lắp camera, tất cả các tuyến đường nhận x làm điểm nút sẽ đều được giám sát. Mỗi tuyến đường có một độ nguy hiểm riêng, đặc trưng cho khả năng xảy ra cướp. Rủi ro của một cách lắp camera là độ nguy hiểm lớn nhất trong số các độ nguy hiểm của các tuyến đường không được giám sát (hoặc bằng 0 nếu tất cả các tuyến đường đều được giám sát)

Hãy giúp bộ giao thông tìm một cách lắp camera thỏa mãn yêu cầu của nhà nước, tức là k tuyến đường đặc biệt đều được giám sát, và rủi ro của cách lắp là nhỏ nhất có thể. Lưu ý là một đường có thể nối một thành phố với chính nó và có thể có nhiều đường cùng nối một cặp thành phố; tạm bỏ qua tính bất hợp lý của việc bộ giao thông lại đi ... giám sát tội phạm trong bài này

Dữ liệu vào

- Dòng đầu ghi số thành phố, số tuyến đường và số tuyến đường đặc biệt: $n\ m\ k$
- m dòng tiếp theo, mỗi dòng ghi một tuyến đường: $u\ v\ w$. $0 \leq w \leq 10^9$ là độ nguy hiểm của tuyến đường, cá biệt $w = -1$ cho biết đây là một tuyến đường đặc biệt

Kết quả

- Dòng đầu ghi rủi ro của cách lắp
- Dòng tiếp theo ghi k số là chỉ số của các thành phố được lắp camera

Ví dụ

stdin	stdout
4 6 2 1 2 -1 1 3 1 1 4 2 2 3 3 2 4 4 3 4 -1	1 2 4

Hạn chế

- $1 \leq n, m, k \leq 10^5$

Bài F. CONECT

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Cho n điểm đen và n điểm trắng trên mặt phẳng, không có hai điểm nào trùng nhau và không có ba điểm nào thẳng hàng. Mỗi đỉnh đen cần được nối với một đỉnh trắng bởi một đoạn thẳng. Hãy tìm một cách nối thỏa mãn không có hai đoạn thẳng nào cắt nhau

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n
- n dòng tiếp theo, mỗi dòng chứa tọa độ một điểm đen
- n dòng tiếp theo, mỗi dòng chứa tọa độ một điểm trắng

Kết quả

In ra n dòng, dòng thứ i chứa chỉ số của điểm trắng nối với điểm đen thứ i . Các điểm được đánh số từ 1 đến n theo thứ tự đầu vào. Nếu không tồn tại cách nối thỏa mãn, in ra -1

Ví dụ

stdin	stdout
3	3
0 0	1
0 2	2
2 0	
0 1	
1 0	
1 1	

Hạn chế

- $1 \leq n \leq 200$, các tọa độ có giá trị tuyệt đối không quá 10000
- Có 30% test với $n \leq 9$

Bài G. BRCNT2

File dữ liệu vào: **stdin**
File kết quả: **stdout**
Hạn chế thời gian: 1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng
- Nếu A là biểu thức ngoặc đúng thì (A) , $[A]$, $\{A\}$ đều là các biểu thức ngoặc đúng
- Nếu A là biểu thức ngoặc đúng, B là biểu thức ngoặc đúng thì AB cũng là biểu thức ngoặc đúng

Cho số nguyên dương n . Ta tiến hành sắp xếp các biểu thức ngoặc đúng có độ dài n theo thứ tự từ điển, với ' $(< [< \{ < ' <) <] < \}$ '. Yêu cầu:

- Cho S là một biểu thức ngoặc đúng có độ dài n . Tìm thứ tự của S ở trong dãy đã sắp xếp ở trên

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên dương n
- Dòng thứ hai chứa xâu S

Kết quả

- Dòng đầu chứa số nguyên dương là thứ tự của S . Chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
4 ([])	2
6 ()[{]}	36

Hạn chế

- $1 \leq n \leq 10^7$, n chẵn
- Subtask 0: $n \leq 10$
- Subtask 1: $n \leq 1000$
- Subtask 2: Ràng buộc gốc