

Mục lục

THREEJUG	2
FACILITY	3
CANDIES	4
CNTDIV	5
STRALT	6
TELMOV	8
ALICEADD	10
SQUARE	11
TAXI	12
BCADIF	13

Nộp bài tại: 202.191.56.251:18888/WinterCamp2020

Username: Họ Tên đầy đủ không dấu. Ví dụ: nguyenvanan

Password như username.

**LƯU Ý: MỖI BÀI CHỈ ĐƯỢC SUBMIT ĐÚNG 1 LẦN TRONG 2H
ĐẦU TIÊN CHO ĐẾN KHI THẦY MỞ SUBMIT LẠI.**

Bài 1. THREEJUG

Có 3 bình dung tích A, B, C (lít) với lượng nước ban đầu tương ứng là a, b, c (lít). Mỗi bước được phép đổ đúng d lít từ một bình i sang một bình j khác với điều kiện lượng nước hiện có trong bình i lớn hơn hoặc bằng d và sau khi đổ hết d lít từ bình i sang bình j thì nước trong bình j không bị tràn ra ngoài. Hãy tìm dãy ít nhất các bước đổ nước sao cho lượng nước còn lại ở 1 trong 3 bình đúng bằng T .

Dữ liệu vào

Dữ liệu đầu vào bao gồm các dòng sau:

- Dòng 1: ghi số 8 số nguyên dương A, B, C, a, b, c, d, T ($0 \leq A, B, C, a, b, c, d, T \leq 10^5$)

Kết quả

Dòng duy nhất ghi số ít nhất các bước đổ nước thỏa mãn yêu cầu đặt ra hoặc ghi ra số -1 nếu không có cách thực hiện nào.

Ví dụ

test	answer
20 20 20 3 6 6 3 12	2
20 20 20 2 4 5 3 10	-1
45 45 45 5 10 8 4 21	4

Hạn chế

- 25% số test với $0 \leq A, B, C \leq 100$.
- 35% số test với $100 < A, B, C \leq 1000$.
- 40% test còn lại không có ràng buộc gì thêm.

Bài 2. FACILITY

Một công ty cung cấp dịch vụ cho thuê kho chứa hàng. Công ty nhận được n đơn đặt thuê kho hàng của khách hàng $1, \dots, n$, mỗi đơn thuê của khách hàng i sẽ bao gồm:

- s_i : ngày bắt đầu thuê
- d_i : số ngày cần thuê
- r_i : số tiền khách hàng i thuê phải trả cho công ty

Tại mỗi thời điểm, kho hàng của công ty chỉ có thể phục vụ cho 1 đơn thuê duy nhất, đồng thời khi một khách hàng kết thúc sử dụng kho hàng thì công ty cần có K ngày để bảo trì kho trước khi cho một khách hàng khác thuê: cụ thể, khách thứ nhất kết thúc thuê vào ngày thứ x thì khách thứ hai chỉ có thể thuê sau ngày thứ $x + K$. Hãy giúp công ty lựa chọn các khách để cho thuê sao cho tổng số tiền thu được là lớn nhất.

Dữ liệu vào

Dữ liệu đầu vào bao gồm các dòng sau:

- Dòng 1: ghi 2 số nguyên dương n, K ($1 \leq n \leq 10^6, 1 \leq K \leq 10^9$)
- Dòng thứ $i + 1$ ($i = 1, \dots, n$): ghi 3 số nguyên dương s_i, d_i, r_i ($1 \leq s_i, d_i, r_i \leq 10^9$)

Kết quả

Tổng số tiền lớn nhất thu được.

Ví dụ

test	answer
5 2 8 3 9 5 1 7 9 2 6 4 3 8 1 2 5	21

Giải thích

Công ty sẽ lựa chọn 3 khách cho thuê là khách thứ 1, 2, và 5 với tổng số tiền thu được là $9 + 7 + 5 = 21$.

Hạn chế

- 30% số điểm có $n, K \leq 10$
- 20% số điểm có $n \leq 10^4, K \leq 10, s_i \leq 10^4, d_i \leq 100, r_i \leq 10^3$
- 30% số điểm có $n, K, s_i, d_i, r_i \leq 5 \times 10^4$
- 20% số điểm còn lại không có ràng buộc gì thêm

Bài 3. 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 4. CNTDIV

Cho số nguyên dương N , xét số nguyên dương $T = N \times (N + 1) \times (N + 2)$. Yêu cầu hãy đếm số ước của T^2 mà nhỏ hơn T và không phải ước của T .

Dữ liệu vào

- Dòng đầu tiên chứa số nguyên Q là số bộ dữ liệu.
- Dòng thứ i trong Q dòng tiếp theo mỗi dòng chứa một số nguyên dương N .

Kết quả

- Ghi ra Q dòng tương ứng với kết quả của từng bộ dữ liệu.

Ví dụ

test	answer
1	3
2	

Hạn chế

- 25% số điểm có $N \leq 1000, Q \leq 10$
- 50% số điểm khác có $N \leq 10^6, Q \leq 10^3$
- 25% số điểm còn lại có $N \leq 10^6, Q \leq 10^5$

Bài 5. STRALT

Alice tìm ra thuật toán riêng để nén một chuỗi yêu thích T chỉ bao gồm các chữ cái viết thường của bảng chữ cái tiếng Anh viết liên tiếp nhau. Chuỗi sau khi nén, ký hiệu là S , có thể bao gồm các số, các chữ cái viết thường của bảng chữ cái tiếng Anh, các ký tự $*$, các dấu ngoặc vuông '[' và ']', và các dấu ngoặc tròn '(' và ')

Bob với bản tính tò mò muốn khám phá ra thuật toán và cố gắng giải nén chuỗi S bằng cách thực hiện các phép biến đổi lặp đi lặp lại. Một phép biến đổi có thể thuộc một trong 3 dạng dưới đây, trong đó chuỗi S chỉ gồm các chữ cái được ký hiệu là C :

- Chuỗi S có dạng $n(C)$, trong đó n là số tự nhiên nằm ngay trước dấu ngoặc tròn, được biến đổi thành chuỗi D thu được bằng cách lặp liên tiếp n lần chuỗi C . Ví dụ, với chuỗi $5(ab)$ ta có $n = 5$ và thu được dãy $D = ababababab$.
- Chuỗi S có dạng $[*C]$ được biến đổi thành một chuỗi palindrom (nghĩa là chuỗi đối xứng) có độ dài chẵn, thu được bằng cách ghép chuỗi C với chuỗi ngược của C . Ví dụ, với chuỗi $[*abc]$, chuỗi palindrom thu được có độ dài chẵn là $abccba$.
- Chuỗi S có dạng $[C*]$ được biến đổi thành một chuỗi palindrom có độ dài lẻ, thu được bằng cách ghép dãy C với chuỗi ngược của C mà bỏ đi ký tự đầu tiên. Ví dụ, với chuỗi $[abc*]$, chuỗi palindrom thu được có độ dài lẻ là $abcca$.

Một chuỗi được coi là đã được giải nén nếu nó chỉ bao gồm các chữ cái viết thường của bảng chữ cái tiếng Anh.

Yêu cầu: Cho chuỗi đã nén S , hãy giúp Bob xác định số lần biến đổi thuộc 3 kiểu trên, cùng với chuỗi T ban đầu trước khi nén của chuỗi S .

Dữ liệu vào

Một dòng duy nhất chứa chuỗi S , các ký tự viết liền nhau.

Kết quả

Dòng đầu tiên chứa một số nguyên là số phép biến đổi tìm được. Dòng thứ hai chứa chuỗi T tìm được.

Ví dụ

test	answer	Giải thích
2(a) [*a2(b)]xy[2(c)b*]	5 aaabbbbaxycbccc	2(a) => aa 2(b) => bb [*a2(b)] => [*abb] => abbbba 2(c) => cc [2(c)b*] => [ccb*] => ccbcc
2(ab[cd*])a3(xyz)	3 abdcabdcacxyzxyzxyz	3(xyz) => xyzxyzxyz [cd*] => cdc 2(ab[cd*]) => 2(abcdc) => abdcabdc
abcd	0 abcd	Không cần biến đổi và chuỗi ban đầu T giống hệt với chuỗi nén S .

Hạn chế

- $0 < |S| \leq 10000$; $0 < |T| \leq 100000$;
- $1 < n \leq 1000$;
- Dữ liệu đảm bảo các xâu đầu vào đúng format nằm trong ba dạng mô tả ở trên và không có dạng $[*S*]$;
- Có 30% tổng số điểm của bài ứng với các bộ test mà chỉ có thể dùng phép biến đổi loại 1;
- Có 30% tổng số điểm của bài ứng với các bộ test khác mà chỉ có thể dùng phép biến đổi loại 2 hoặc loại 3.

Bài 6. TELMOV

Cô kỹ sư Alice đang sống ở trong thiên hà VNOI2020. Trong thiên hà này có N hành tinh khác nhau và M kênh vận chuyển hai chiều dạng (x, y, t) cho phép bạn di chuyển từ hành tinh x đến hành tinh y (hoặc ngược lại) trong t giây.

Nhưng Alice nhận thấy phương pháp vận chuyển này rất kém hiệu quả nên đã phát triển một thiết bị cho phép bạn dịch chuyển từ hành tinh x đến bất kỳ hành tinh y nào khác trong P giây với điều kiện bạn có thể đến hành tinh y đó từ hành tinh x chỉ sử dụng tối đa L kênh vận chuyển.

Thiết bị này hiện mới là bản thử nghiệm nên không thể được sử dụng quá K lần. Alice đang ở hành tinh 1 và muốn biết thời gian tối thiểu để đến hành tinh N .

Yêu cầu: Viết chương trình tính thời gian tối thiểu cần thiết để đến được hành tinh N bắt đầu từ hành tinh 1.

Dữ liệu vào

Dòng đầu tiên chứa 5 giá trị N, M, P, L, K cách nhau một dấu cách.

Mỗi dòng trong số M dòng sau chứa 3 giá trị X_i, Y_i, T_i mô tả một kênh vận chuyển. Dữ liệu đảm bảo có nhiều nhất một kênh giữa hai hành tinh.

Kết quả

Kết quả ghi ra một giá trị duy nhất là thời gian tối thiểu cần thiết để đến hành tinh N bắt đầu từ hành tinh 1. Dữ liệu đảm bảo luôn có đáp án.

Ví dụ

test	answer	Giải thích
6 7 3 2 1 1 2 2 1 3 5 2 3 4 2 4 23 3 4 6 5 4 7 5 6 9	14	Thiết bị có thể được sử dụng một lần. Để đến hành tinh 6 trong thời gian tối thiểu, chúng ta sẽ đi qua kênh 1 -> 2 sau đó sẽ dịch chuyển đến hành tinh 5 từ đó sẽ đi qua kênh 5 -> 6. Chi phí cuối cùng là $2 + 3$ (dịch chuyển bởi thiết bị) $+ 9 = 14$.
6 7 3 2 0 1 2 2 1 3 5 2 3 4 2 4 23 3 4 6 5 4 7 5 6 9	27	Thiết bị hoàn toàn không thể sử dụng được. Để đến hành tinh 6 từ hành tinh 1 trong thời gian tối thiểu, cần đi qua các kênh theo thứ tự 1 -> 3 -> 4 -> 5 -> 6 và với thời gian $5 + 6 + 7 + 9 = 27$ giây.

Hạn chế

- $1 < N, \leq 10000, 1 < M \leq 20000$;
- $0 \leq L, K \leq 10$;
- $1 < T_i, P \leq 100000$;

- $1 < X_i, Y_i \leq N$;
- 24% số điểm ứng với các test có $K = 0$ và tất cả các kênh vận chuyển đều có $T_i = 1$;
- 16% số điểm ứng với các test khác có $K = 0$;
- 16% số điểm ứng với các test khác đảm bảo $N \leq 300$;

Bài 7. ALICEADD

LƯU Ý: BÀI CHỈ ĐƯỢC PHÉP SUBMIT DUY NHẤT 1 LẦN

Alice có a cái kẹo, Bob cho Alice thêm b cái kẹo. Hỏi Alice có tất cả bao nhiêu cái kẹo?

Dữ liệu vào

- Dòng đầu chứa một số nguyên không âm T là số bộ dữ liệu ($T \leq 10$).
- Mỗi dòng trong số T dòng tiếp theo chứa hai số nguyên không âm a và b cách nhau bởi dấu cách ($a, b \leq 10^{19}$).

Kết quả

Gồm T dòng là kết quả cho T bộ dữ liệu theo thứ tự đầu vào.

Ví dụ

test	answer
2	8
3 5	5
4 1	

Bài 8. SQUARE

Xét dãy số sau: $0, 0 + 1, 0 + 1 + 3, 0 + 1 + 3 + 5, \dots, 0 + 1 + 3 + \dots + (2n - 1), \dots$. Đây là dãy được tạo bởi tổng vài số tự nhiên lẻ đầu tiên và các số hạng của dãy đều là số chính phương (tức là bình phương của một số nguyên): $0, 1, 4, 9, \dots, n^2, \dots$.

Tổng quát hóa dãy này bằng cách thay số 0 ở đầu bởi một số nguyên k , như vậy ta được dãy: $k, k + 1, k + 1 + 3, k + 1 + 3 + 5, \dots, k + 1 + 3 + \dots + (2n - 1), \dots$. Tuy nhiên khác với trường hợp $k = 0$ ở trên, dãy này chỉ có một vài số hạng là số chính phương.

Yêu cầu: Cho trước số nguyên k , cần tìm số nguyên không âm nhỏ nhất sao cho bình phương của nó xuất hiện trong dãy số trên.

Dữ liệu vào

Một dòng chứa số nguyên duy nhất là k ($-10^{12} \leq k \leq 10^{12}$).

Kết quả

Ghi ra một số nguyên không âm duy nhất sao cho bình phương của nó xuất hiện trong dãy số trên. Nếu trong dãy không có số chính phương nào, hay ghi ra xâu "none".

Hạn chế

- Có 8% số lượng test tương ứng 8% số điểm thỏa mãn $0 \leq k \leq 1000$;
- Có 12% số lượng test tương ứng 12% số điểm thỏa mãn $0 \leq k \leq 10^5$;
- Có 20% số lượng test tương ứng 20% số điểm thỏa mãn $0 \leq k \leq 10^{12}$;
- Có 8% số lượng test tương ứng 8% số điểm thỏa mãn $-1000 \leq k \leq 1000$;
- Có 12% số lượng test tương ứng 12% số điểm thỏa mãn $-10^5 \leq k \leq 10^5$;
- 40% số lượng test còn lại tương ứng 40% số điểm thỏa mãn $-10^{12} \leq k \leq 10^{12}$.

Ví dụ

test	answer
0	0
-5	2
2	none

Bài 9. TAXI

Crab vừa rộng mô hình dịch vụ sang chuyển phát hàng hóa khi xe đang rảnh. Có n gói hàng, gói thứ i muốn chuyển từ vị trí i đến vị trí $i + n$. Cần lập lịch cho xe xuất phát từ vị trí 0, chuyển hết các gói hàng và quay lại vị trí xuất phát. Sức chứa của xe là đủ lớn, do đó gói hàng thứ i sẽ được chuyển nếu ít nhất một lần, lộ trình của xe có đi qua i trước khi đi qua $i + n$. Ví dụ với $n = 3$, lộ trình sau là thỏa mãn: $0 - 1 - 2 - 1 - 5 - 3 - 6 - 4 - 0$

Cho biết độ dài tuyến đường đi lại giữa mọi cặp vị trí, hãy tìm lộ trình của taxi có tổng độ dài các tuyến đường đi qua là nhỏ nhất. Lưu ý, các tuyến đường trong thành phố là đường một chiều nên khoảng cách từ x đến y có thể khác với khoảng cách từ y đến x , và có thể đường đi ngắn nhất x và y không phải là đường đi trực tiếp giữa chúng. Nếu có nhiều lộ trình thỏa mãn có cùng độ dài nhỏ nhất, in ra một lộ trình bất kỳ

Dữ liệu vào

- Dòng 1: n
- Tiếp theo là $2n + 1$ dòng, số thứ j trên dòng i là $c_{i,j}$: độ dài tuyến đường nối i với j

Kết quả

- Dòng đầu tiên chứa tổng độ dài của lộ trình tìm được
- Dòng tiếp theo chứa số vị trí sẽ đi qua
- Dòng tiếp theo ghi danh sách các vị trí sẽ đi qua theo thứ tự trong lộ trình

Ví dụ

test	answer
3	12
0 4 2 3 5 4 4	9
4 0 7 5 2 3 1	0 2 5 2 3 1 4 6 0
3 2 0 1 2 1 9	
2 3 5 0 9 8 3	
2 1 4 6 0 9 1	
9 8 1 4 2 0 8	
1 2 3 2 5 4 0	

Hạn chế

- $1 \leq n \leq 10$. $1 \leq c_{i,j} \leq 1000$
- Subtask 1: $n \leq 5$
- Subtask 2: $c_{i,j} + c_{j,k} \geq c_{i,k} \forall 0 \leq i, j, k \leq 2n$
- Subtask 3: Ràng buộc gốc

Bài 10. BCADIF

At the beginning of the semester, the head of a computer science department D have to assign courses to teachers in a balanced way. The department D has m teachers $T = \{1, 2, \dots, m\}$ and n courses $C = \{1, 2, \dots, n\}$. Each teacher $t \in T$ has a preference list which is a list of courses he/she can teach depending on his/her specialization. We known a list of pairs of conflicting two courses that cannot be assigned to the same teacher as these courses have been already scheduled in the same slot of the timetable. The load of a teacher is the number of courses assigned to her/him. How to assign n courses to m teacher such that each course assigned to a teacher is in his/her preference list, no two conflicting courses are assigned to the same teacher, and the maximal load is minimal.

Dữ liệu vào

The input consists of following lines

- Line 1: contains two integer m and n ($1 \leq m \leq 15, 1 \leq n \leq 30$)
- Line $i+1$: contains an positive integer k and k positive integers indicating the courses that teacher i can teach ($\forall i = 1, \dots, m$)
- Line $m + 2$: contains an integer q
- Line $i + m + 2$: contains two integer i and j indicating two conflicting courses ($\forall i = 1, \dots, q$)

Kết quả

The output contains a unique number which is the maximal load of the teachers in the solution found and the value -1 if not solution found.

Ví dụ

test	answer
4 12 5 1 3 5 10 12 5 9 3 4 8 12 6 1 2 3 4 9 7 7 1 2 3 5 6 10 11 25 1 2 1 3 1 5 2 4 2 5 2 6 3 5 3 7 3 10 4 6 4 9 5 6 5 7 5 8 6 8 6 9 7 8 7 10 7 11 8 9 8 11 8 12 9 12 10 11 11 12	3