

Bài 1: Lucky Number

*Giới hạn thời gian: 1 second
Giới hạn bộ nhớ: 256 megabytes
Đầu vào: vào ra chuẩn
Đầu ra: vào ra chuẩn*

Alice chuẩn bị tham dự *Coding Challenge* trong chuỗi sự kiện Devfest 2021 do GDG Hà Nội và GDSC HUST đồng tổ chức. Là một người theo hệ tâm linh, trước mỗi kì thi Alice đều chọn ngẫu nhiên cho mình 3 số a, b, c . Nếu trong 3 số a, b, c tồn tại **ít nhất** một số bằng **2021**, thì bộ ba số đó là bộ ba may mắn và nó sẽ đem lại may mắn cho Alice trong cuộc thi.

Nhiệm vụ của bạn là xác định xem Alice có chọn được bộ ba số may mắn hay không? Nếu là bộ ba số may mắn, in ra “YES”, ngược lại in ra “NO”.

Input:

- Dòng đầu chứa số nguyên T – tương ứng với số bộ test ($T \leq 1000$)
- T dòng tiếp theo, mỗi dòng chứa 3 số nguyên a, b, c ($1 \leq a, b, c \leq 10^9$)

Output:

- Gồm T dòng, dòng i tương ứng với kết quả của bộ test thứ i .

Examples:

Input	Output
2	NO
10 20 30	YES
20 2021 21	

Bài 2: Palindrome

*Giới hạn thời gian: 1 second
Giới hạn bộ nhớ: 512 megabytes
Đầu vào: vào ra chuẩn
Đầu ra: vào ra chuẩn*

Đến ngày thi *Coding Challenge*, Alice được ban tổ chức phát cho một account bất kì. Alice rất thích những account là một chuỗi đối xứng. Alice đặt ra câu hỏi là cần thay đổi **ít nhất** bao nhiêu kí tự để account của mình là **một chuỗi đối xứng**.

Palindrome hay còn gọi là chuỗi đối xứng, chuỗi đối gương là tên gọi của những chuỗi kí tự mà khi viết từ phải qua trái hay từ trái qua phải thì chuỗi đó không thay đổi.

Input:

- Một chuỗi S duy nhất chứa account của Alice (Độ dài của chuỗi S nằm trong khoảng từ 1 đến 1000).

Output:

- Một số nguyên duy nhất chứa số kí tự **ít nhất** cần thay đổi để S là chuỗi đối xứng.

Examples:

Input	Output
devfest	3
abcaa	1

Bài 3: Find

Giới hạn thời gian: 1 second
 Giới hạn bộ nhớ: 512 megabytes
 Đầu vào: vào ra chuẩn
 Đầu ra: vào ra chuẩn

Trước khi đến với *Coding Challenge*, công việc hàng ngày của Alice là đi đốn cây lấy gỗ trong khu rừng của mình. Khu rừng của Alice là một hàng cây gồm N cây, nằm trên một đường thẳng, cây thứ i có chiều cao là h_i mét. Để đốn gỗ, Alice sẽ chọn một đoạn liên tiếp các cây và chặt hết tất cả các cây đó. Alice tự hỏi có bao nhiêu cách chọn một đoạn liên tiếp các cây để đốn sao cho thu được **ít nhất** M mét gỗ.

Input:

- Dòng đầu chứa 2 số nguyên dương N và M ($1 \leq N \leq 10^5, 1 \leq M \leq 10^{10}$) – tương ứng với số lượng cây và số lượng mét gỗ cần phải đốn.
- Dòng thứ hai chứa N số nguyên dương $h_1, h_2, h_3, \dots, h_N$ ($1 \leq h_i \leq 10^5$) tương ứng với chiều cao của từng cây.

Output:

- Số cách chọn liên tiếp các cây để đốn sao cho thu được **ít nhất** M mét gỗ..

Examples:

Input	Output	Note
4 10 5 1 3 6	2	Có hai cách chọn thoả mãn: - $h[1..4]$ với tổng bằng 15; - $h[2..4]$ với tổng bằng 10.
3 5 4 4 4	3	Có 3 cách chọn thoả mãn: - $h[1..2]$ với tổng bằng 8; - $h[1..3]$ với tổng bằng 12; - $h[2..3]$ với tổng bằng 8.

Subtask:

- Task 1 (30% số điểm): $N \leq 100$.
- Task 2 (30% số điểm): $N \leq 5000$
- Task 3 (40% số điểm): Không có ràng buộc gì thêm.

Bài 4: Cutting Tree

Giới hạn thời gian: 3 second
Giới hạn bộ nhớ: 512 megabytes
Đầu vào: vào ra chuẩn
Đầu ra: vào ra chuẩn

Lần này, Alice tiếp tục phải đốn cây nhưng mà sẽ đốn theo một cách khác. Khu rừng của Alice là một hàng cây gồm N cây, nằm trên một đường thẳng, cây thứ i có chiều cao là h_i mét. Mỗi khi đốn gỗ, Alice sẽ đặt một máy cưa ở một độ cao X nguyên dương, và sẽ di chuyển dọc từ đầu đến cuối. Khi đó, với những cây mà có chiều cao lớn hơn X , thì sẽ bị cắt đi một đoạn là $h_i - X$ mét. Ví dụ, có 5 cây với chiều cao lần lượt là 20,15,10,17,16 nếu đặt cưa ở độ cao $X = 15$, thì sau khi di chuyển dọc từ đầu đến cuối, ta sẽ thu được 5 mét ở cây thứ nhất, 2 mét ở cây thứ 4 và 1 mét ở cây thứ 5. Tổng số gỗ thu được là $5 + 2 + 1 = 8$ (mét). Khi đó, chiều cao còn lại của các cây lần lượt là 15 15 10 15 15.

Hôm nay, sau khi tham dự *Coding Challenge*, Alice có nhiệm vụ phải đốn M mét gỗ và chỉ được đặt **một lần cưa**. Alice rất yêu môi trường, nên bạn hãy giúp Alice tìm độ cao X **lớn nhất** để đặt máy cưa sao cho sau khi di chuyển từ đầu đến cuối, số gỗ thu được phải lớn hơn hoặc bằng M .

Input:

- Dòng đầu chứa 2 số nguyên dương N và M – tương ứng với số lượng cây và số lượng mét gỗ cần phải đốn
- Dòng thứ hai chứa N số nguyên dương $h_1, h_2, h_3, \dots, h_N$ tương ứng với chiều cao của từng cây.

Constraints:

- $1 \leq N \leq 1000000$
- $1 \leq M \leq 2000000000$
- $1 \leq h_i \leq 1000000000$
- Tổng chiều cao của tất cả các cây $\geq M$.

Output:

- Chiều cao X lớn nhất để đặt máy cưa thỏa mãn yêu cầu.

Examples:

Input	Output	Note
5 6 17 15 10 20 16	15	Chiều cao lớn nhất để đặt máy cưa là 15. Khi đó sẽ thu được: $2 + 0 + 0 + 5 + 1 = 8$ (mét)
5 21 17 15 10 20 16	11	Chiều cao lớn nhất để đặt máy cưa là 11. Khi đó sẽ thu được: $6 + 4 + 0 + 9 + 5 = 24$ (mét)

Subtask:

- Task 1(30% số điểm): $1 \leq N, h_i \leq 100, M \leq 1000$
- Task 2(30% số điểm): $1 \leq N, h_i \leq 1000, M \leq 10000$
- Task 3(40% số điểm): Không có ràng buộc gì thêm.

Bài 5: Number

*Giới hạn thời gian: 3 second
Giới hạn bộ nhớ: 512 megabytes
Đầu vào: vào ra chuẩn
Đầu ra: vào ra chuẩn*

Trong một lần trồng rừng, Alice đã vô tình nhặt được một cuộn giấy có lời tiên tri cổ xưa. Alice không biết một ngôn ngữ cổ đại nào và do đó Alice không thể hiểu được lời tiên tri. Nhưng anh ấy biết các chữ số.

Một đoạn của lời tiên tri là một dãy gồm có n chữ số, chữ số đầu tiên không bắt đầu bởi số 0. Alice nghĩ dãy số đó là danh sách của một số năm đặc biệt. Thật khó để nhìn thấy bất kỳ dấu phẩy hoặc dấu cách nào, vì có thể người cổ đại đã không sử dụng chúng. Bây giờ Alice tự hỏi những năm được liệt kê ở đó.

Alice đã đưa ra 3 giả định về dãy số:

- Các năm sẽ được liệt kê theo thứ tự tăng dần;
- Mỗi năm đề là số nguyên dương;
- Các năm không bắt đầu bằng chữ số 0.

Alice sẽ xem xét tất cả các cách để có thể chia dãy thành các số (năm), thỏa mãn 3 điều kiện ở trên. Và Alice tự hỏi là có bao nhiêu cách như vậy.

Vì con số này rất lớn, nên kết quả sẽ được tính theo mô-đun của $10^9 + 7$ (nghĩa là kết quả sẽ là số dư của tổng số cách khi chia cho $10^9 + 7$).

Input:

- Dòng đầu tiên chứa số nguyên dương n – số lượng chữ số của dãy ($1 \leq n \leq 5000$)
- Dòng thứ hai chứa một xâu gồm các chữ số có độ dài bằng n . Dữ liệu đảm bảo chữ số đầu tiên khác '0'.

Output:

- In ra số cách có thể chia dãy, kết quả được tính theo mô-đun của $10^9 + 7$.

Examples:

Input	Output	Note
6 234545	8	<p>Có 8 cách chia dãy như sau:</p> <ul style="list-style-type: none"> - “234545” = “234545” - “234545” = “2” + “34545” - “234545” = “23” + “4545” - “234545” = “234” + “545” - “234545” = “2” + “34” + “545” - “234545” = “2” + “3” + “4545” - “234545” = “2” + “3” + “4” + “545” - “234545” = “2” + “3” + “4” + “5” + “45” <p>Lưu ý, ta không để chia thành “234545” = “23” + “45” + “45” bởi vì dãy được chia ra phải là một dãy tăng nghiêm ngặt.</p>
8 20212022	5	<p>Có 5 cách để chia dãy như sau:</p> <ul style="list-style-type: none"> - “20212022” = “20212022” - “20212022” = “20” + “212022” - “20212022” = “202” + “12022” - “20212022” = “20” + “21” + “2022” - “20212022” = “2021” + “2022”

Subtask:

- Sub 1 (30% số điểm) : $n \leq 15$.
- Sub 2 (20% số điểm): $n \leq 100$.
- Sub 3 (50% số điểm): không có ràng buộc gì thêm.

Bài 6: Sum

Giới hạn thời gian: 2 second
Giới hạn bộ nhớ: 512 megabytes
Đầu vào: vào ra chuẩn
Đầu ra: vào ra chuẩn

Alice đang làm leader của một team gồm có N người đánh số từ 1 đến N . Đối với một nhiệm vụ cụ thể, Alice sẽ chọn ra một tập hợp con không rỗng trong N người để làm nhiệm vụ đó. Chi phí để có x người cho nhiệm vụ sẽ là x^k .

Alice đặt ra một câu hỏi: Liệu có thể tính tổng chi phí của tất cả các khả năng chia nhóm không?

Input:

- Chứa 2 số nguyên dương N ($1 \leq N \leq 10^9$) đại diện cho tổng số người và số nguyên dương k ($1 \leq k \leq 5000$)

Output:

- Tổng chi phí của tất cả các cách chia nhóm. Vì kết quả khá lớn, in ra kết quả theo mô-đun khi chia cho $10^9 + 7$ (phần dư khi chia cho $10^9 + 7$)

Examples:

Input	Output	Note
1 1	1	Chỉ có 1 tập con không rỗng là : $\{1\}$ với chi phí $1^1 = 1$
3 2	24	<p>Có 7 tập con không rỗng có thể chọn là:</p> <ul style="list-style-type: none"> $\{1\}$ với chi phí $1^2 = 1$ $\{1,2\}$ với chi phí $2^2 = 4$ $\{1,3\}$ với chi phí $2^2 = 4$ $\{1,2,3\}$ với chi phí $3^2 = 9$ $\{2\}$ với chi phí $1^2 = 1$ $\{2, 3\}$ với chi phí $2^2 = 4$ $\{3\}$ với chi phí $1^2 = 1$ <p>Tổng chi phí của 7 cách là: $1 + 4 + 4 + 9 + 1 + 4 + 1 = 24$.</p>

Subtask:

- Task 1(10% số điểm) : $n \leq 20$
- Task 2(20% số điểm): $n \leq 5000$
- Task 3(70% số điểm): không có ràng buộc gì thêm

Bài 7: Diameter

*Giới hạn thời gian: 7 second
Giới hạn bộ nhớ: 1024 megabytes
Đầu vào: vào ra chuẩn
Đầu ra: vào ra chuẩn*

Bạn có một đồ thị vô hướng có trọng số gồm N đỉnh $N - 1$ cạnh thỏa mãn hai đỉnh bất kì có thể đi đến nhau qua một số cạnh. Người ta gọi đồ thị thỏa mãn điều kiện trên là cây.

Đường đi giữa hai đỉnh được tính là tổng trọng số của các cạnh nằm trên đường đi đó.

Gọi $f(x, y)$ là đường đi ngắn nhất giữa hai đỉnh x và y trên đồ thị, khi đó định nghĩa “đường kính của cây” được tính bằng $\max_{1 \leq u, v \leq N} f(u, v)$ – đường đi lớn nhất giữ 2 đỉnh bất kỳ trên cây.

Bạn cần phải thực hiện Q truy vấn, mỗi truy vấn thực hiện thay đổi trọng số của đúng một cạnh. Nhiệm vụ của bạn là tính đường kính của cây sau mỗi truy vấn được cập nhật.

Input:

- Dòng đầu chứa ba số nguyên N, Q và W ($2 \leq N \leq 10^5; 1 \leq Q \leq 10^5; 1 \leq W \leq 2 \cdot 10^{13}$) – lần lượt là số đỉnh của cây, số lượng các truy vấn và giới hạn trọng số các cạnh trên cây.
- $N - 1$ dòng tiếp theo, dòng thứ i gồm ba số nguyên a_i, b_i, c_i ($1 \leq a_i, b_i \leq N; 0 \leq c_i < W$) – biểu diễn cạnh của cây nối hai đỉnh a_i và b_i với trọng số c_i .
- Q dòng cuối, dòng thứ j gồm hai số nguyên d_j và e_j ($0 \leq d_j < N - 1; 0 \leq e_j < W$). Sau đó thực hiện như sau:
 - $d'_j = (d_j + last) \bmod (N - 1)$
 - $e'_j = (e_j + last) \bmod W$

trong trường hợp này, $last$ là kết quả của truy vấn cuối cùng (ban đầu $last = 0$). Cặp (d'_j, e'_j) chính là việc yêu cầu bạn cần phải thay đổi trọng số cạnh thứ $d'_j + 1$ trong input thành e'_j .

Output

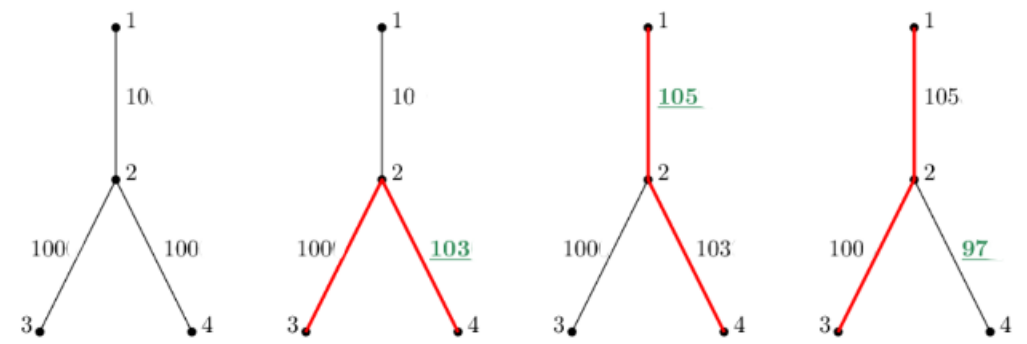
- Gồm Q dòng. Dòng thứ i trả lời cho truy vấn thứ i là đường kính của cây ở thời điểm đó. Lưu ý, truy vấn thứ i có thể **ảnh hưởng hoàn toàn** đến các truy vấn sau, các truy vấn không độc lập mà diễn ra tuần tự.

Ví dụ

Input	Output
4 3 200	203
1 2 10	208
2 3 100	205
2 4 100	
2 103	
1 102	
1 89	
10 10 10000	16531
1 9 1224	16531
5 6 1623	16531
10 5 1623	17462
2 6 825	23825
10 1 5231	23825
5 3 7426	19593
8 3 1027	19593
4 10 1428	18371
7 10 423	18371
8 2035	
5 6239	
5 4126	
7 1826	
0 524	
6 515	

3 300	
8 3526	
5 3310	
8 3622	

Giải thích testcase 1 bằng hình ảnh dưới đây



Giới hạn

- Subtask 1 (18%): $N, Q \leq 100$; $W \leq 10^4$.
- Subtask 2 (8%): $N, Q \leq 5000$; $W \leq 10^4$.
- Subtask 3 (21%): $W \leq 10^4$ và các cạnh trên cây có dạng $\{1, i\}$ (dạng hình các đỉnh bao quanh đỉnh 1).
- Subtask 4 (17%): $W \leq 10^4$ và các cạnh trên cây có dạng $\{i, 2i\}$ và $\{i, 2i + 1\}$ (dạng cây nhị phân nếu xem gốc là 1).
- Subtask 5 (24%): Dữ liệu đảm bảo rằng sau khi update, đường kính của cây sẽ luôn đi qua đỉnh 1.
- Subtask 6 (12%): Không giới hạn gì thêm.