

Buy

Bình có số tiền N . Giá một cái bút là A và một quyển vở là B . Cậu ta có thể mua số lượng bút và vở bất kỳ không âm. Em hãy tìm hiểu xem, Bình có thể mua được một số lượng bút và vở sao cho Bình tiêu hết số tiền N hay không.

Nói cách khác, bạn cần tìm hai số nguyên không âm x và y sao cho Bình có thể mua x cái bút và y cuốn vở và $x \times A + y \times B = N$ hoặc cho biết điều đó là không thể.

Đầu vào

Gồm một dòng chứa lần lượt ba số nguyên N, A, B cách nhau bởi dấu cách.

Ràng buộc

- $1 \leq N, A, B \leq 10^7$

Đầu ra

Nếu Bình không thể mua bút và vở theo cách mà để tiêu hết chính xác số tiền N thì in ra thông báo NO. Ngược lại thì, in ra dòng đầu tiên thông báo YES. Dòng thứ hai in hai số nguyên không âm x và y tương ứng là số lượng bút và vở mà Bình có thể mua để tiêu hết số tiền N , tức là $x \times A + y \times B = N$. Nếu có nhiều câu trả lời đúng thì chỉ cần in ra một câu trả lời bất kỳ trong số đó.

Ví dụ

Đầu vào	Đầu ra
7 2 3	YES 2 1
100 25 10	YES 0 10
15 4 8	NO

Giải thích

- Ví dụ 1, Bình có thể mua 2 bút và 1 vở. Cậu ta sẽ tiêu hết số tiền $2 \times 2 + 1 \times 3 = 7$.
- Ví dụ 2, Bình có thể chỉ tiêu hết số tiền 100 theo nhiều cách, ví dụ: mua 2 bút và 5 vở, hoặc mua 4 bút và 0 vở hoặc mua 0 bút và 10 vở.
- Ví dụ 3, không có cách nào để chỉ tiêu hết số tiền 15.

Rounding

An có một số nguyên N không âm. Cậu ta muốn làm tròn nó đến số nguyên gần nhất mà chia hết cho 10. Nếu N chia hết cho 10, An coi nó như đã được làm tròn.

Ví dụ: nếu $N = 4722$ câu trả lời là 4720. Nếu $N = 5$ thì An có thể làm tròn nó thành 0 hoặc 10. Cả hai cách đều đúng.

Cho biết N em hãy lập trình tìm số nguyên mà An muốn làm tròn.

Đầu vào

Gồm một dòng chứa số nguyên N là số mà An đang có.

Ràng buộc

- $0 \leq N \leq 10^9$

Đầu ra

In ra kết quả làm tròn của N theo cách An muốn. Trong một số trường hợp, câu trả lời không duy nhất, chương trình của bạn có thể in ra bất kỳ câu trả lời nào đúng.

Ví dụ

Đầu vào	Đầu ra
5	0
113	110
1000000000	1000000000
5432359	5432360

Giải thích

Ví dụ 1, $N=5$, các số nguyên gần N nhất chia hết cho 10 là 0 và 10, chương trình của bạn có thể in ra 0 hoặc 10 đều được coi là đúng.

Water

Giờ ra chơi, trên sân trường có N học sinh, các học sinh được gán số hiệu từ 1 đến N . Sau giờ ra chơi, các em muốn uống nước. Chỉ có một thùng nước uống duy nhất nên mỗi thời điểm chỉ có một học sinh được sử dụng, vì vậy các học sinh phải xếp hàng để uống nước.

Học sinh thứ i đến và xếp vào cuối hàng đợi vào giây thứ $L[i]$. Nếu có nhiều học sinh đến xếp hàng vào cùng một thời điểm, thì học sinh có số hiệu lớn hơn sẽ đứng sau học sinh có số hiệu nhỏ hơn. Các học sinh trong hàng đợi cư xử như sau: học sinh đứng đầu hàng sử dụng bình nước trong

đứng 1 giây và rời khỏi hàng với cốc nước của mình; các học sinh đứng sau này chờ người đứng trước lấy nước. Nếu đến giây thứ $R[i]$ mà học sinh thứ i vẫn chưa đến lượt lấy nước thì bạn ấy sẽ rời khỏi hàng đợi mà không uống nước nữa.

Với mỗi học sinh, em hãy xác định số giây mà bạn ấy sẽ sử dụng bình nước (nếu bạn ấy vẫn chờ được) hoặc cho biết bạn này không chờ được đến lượt mình và đã bỏ hàng.

Đầu vào

Dòng đầu tiên của đầu vào chứa một số nguyên T là số Bộ dữ liệu vào cần kiểm tra. Mỗi Bộ dữ liệu vào gồm:

- Dòng đầu chứa một số nguyên N là số lượng học sinh.
- Dòng thứ i trong N dòng sau chứa hai số nguyên $L[i]$, $R[i]$ cho biết thời điểm học sinh thứ i vào cuối hàng đợi và thời điểm học sinh này rời khỏi hàng đợi nếu cậu ta vẫn không thể lấy được nước. Đầu vào đảm bảo $L[i - 1] \leq L[i]$ ($\forall i = 2..N$).

Ràng buộc

- $1 \leq T, N \leq 1000$, tổng giá trị N trong tất cả các bộ dữ liệu vào không vượt quá 1000.
- $1 \leq L[i] \leq R[i] \leq 5000, i = 1..N$

Đầu ra

Ứng với mỗi Bộ dữ liệu vào, chương trình của bạn cần in ra một dòng chứa N số nguyên, số thứ i là số giây mà học sinh thứ i lấy được nước và ra khỏi hàng, hoặc in ra 0 nếu cậu ta rời hàng mà không uống nước.

Ví dụ

Đầu vào	Đầu ra
2	1 2
2	1 0 2
1 3	
1 4	
3	
1 5	
1 1	
2 3	

Giải thích

Ví dụ trên có 2 bộ dữ liệu vào ($T = 2$):

- Với bộ dữ liệu vào đầu tiên: Trong giây thứ 1, học sinh thứ 1 và 2 đến xếp hàng và học sinh thứ 1 được uống nước vào giây 1, học sinh thứ 2 được uống nước vào giây thứ 2.
- Với bộ dữ liệu vào thứ hai: Trong giây thứ 1, học sinh 1 và 2 đến xếp hàng, học sinh 1 được uống nước vào giây thứ 1 và học sinh 2 không đợi được uống nước đã bỏ hàng. Trong giây thứ 2, học sinh thứ 3 đến và uống nước vào giây thứ 2.

kstring

Một chuỗi được gọi là *k-string* là chuỗi ký tự có dạng lặp lại k lần một chuỗi nào đó. Ví dụ, chuỗi "hoahoa" được gọi là chuỗi 2-string vì nó là kết quả của việc viết lặp lại 2 lần chuỗi "hoa". Chuỗi "abababab" vừa là chuỗi 2-tring và vừa là chuỗi 4-string, còn chuỗi "abc" chỉ là chuỗi 1-string. Tất nhiên một chuỗi bất kỳ luôn là chuỗi 1-string.

Cho số nguyên dương k và chuỗi ký tự s , hãy tìm cách sắp xếp lại thứ tự các ký tự trong s để chuỗi kết quả là một chuỗi *k-string*. Nếu không thể thực hiện được thì in ra số -1.

Đầu vào

- Dòng đầu chứa số nguyên k ($1 \leq k \leq 10^3$).
- Dòng thứ hai chứa chuỗi ký tự s khác rỗng và chỉ gồm các chữ cái tiếng Anh in thường (chuỗi s có độ dài không quá 10^4).

Đầu ra

Nếu có nhiều đáp án thỏa mãn yêu cầu bài toán thì chỉ cần in ra một đáp án bất kỳ. Ngược lại, nếu không thực hiện được thì in ra số -1.

Ràng buộc

- Có 60% số test ứng với 60% điểm với chuỗi s có độ dài không quá 200.

Ví dụ

Đầu vào	Đầu ra
2 aabb	abab
2 abaa	-1