**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 1. Xây dựng mảng cộng dồn**

Cho mảng số nguyên A[] gồm N phần tử, mảng cộng dồn của A[] là mảng F[] với tính chất F[i] lưu tổng các phần tử từ chỉ số 0 tới chỉ số i của mảng A[]. Bạn hãy xây dựng mảng cộng dồn F[]

**Input Format**

* Dòng 1 là N
* Dòng 2 là N số nguyên

**Constraints**

* 1<=N<=10^6
* 1<=A[i]<=10^9

**Output Format**

In ra mảng cộng dồn

**Sample Input 0**

6

8 7 6 5 6 6

**Sample Output 0**

8 15 21 26 32 38

**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 2. Truy vấn tổng tĩnh**

Cho mảng số nguyên A[] gồm N phần tử, có Q truy vấn, mỗi truy vấn là 2 số L, R bạn hãy tính tổng các số từ chỉ số L tới chỉ số R của mảng.

**Input Format**

* Dòng 1 là N và Q
* Dòng 2 là N số nguyên
* Q dòng tiếp theo mỗi dòng là 1 truy vấn

**Constraints**

* 1<=N,Q<=10^6
* 1<=A[i]<=10^9
* 0<=L<=R<=N-1

**Output Format**

In ra kết quả của mỗi truy vấn trên 1 dòng

**Sample Input 0**

11 3

6 5 9 7 7 6 7 7 5 9 7

1 9

1 10

0 8

**Sample Output 0**

62

69

59

**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 3. Thu hoạch cà rốt**

Tèo hiện tại đã bỏ làm lập trình viên và trở về về quê trồng rau nuôi cá, anh ta có một mảnh vườn hình chữ nhật có kích thước NxM. Anh ta chia vườn của mình thành NXM ô vuông và trồng vào đó một cây cà rốt, tới vụ thu hoạch có những cây cà rốt bị chết và có những cây cà rốt có củ, anh ta muốn biết với mỗi mảnh vườn hình chữ nhật bắt đầu từ hàng x1 tới hàng x2 và từ cột y1 tới cột y2 thì số cà rốt thu hoạch được là bao nhiêu.

Biết rằng mảnh vườn được mô tả bởi một ma trận nhị phân, 0 tương ứng với cây cà rốt chất và 1 tương ứng với cây cà rốt có củ.

**Input Format**

* Dòng 1 là N và M
* N dòng tiếp theo mỗi dòng M số mô tả mảnh vườn
* Dòng tiếp theo là Q : số lượng truy vấn
* Q dòng tiếp theo mỗi dòng gồm 4 số : x1, x2, y1, y2

**Constraints**

* 1<=N,M<=1000
* 1<=Q<=10^5
* 1<=x1,x2<=N, 1<=y1,y2<=M

**Output Format**

Đối với mỗi truy vấn in ra số lượng cà rốt thu hoạch

**Sample Input 0**

8 8

1 1 0 0 0 1 1 1

1 0 0 0 1 1 1 1

0 1 1 1 0 0 1 0

1 1 1 1 0 1 1 0

1 1 1 1 1 0 1 1

1 0 0 1 0 1 0 1

0 0 0 0 1 0 1 0

1 1 0 0 0 1 0 1

3

2 3 1 7

2 2 2 7

1 2 1 8

**Sample Output 0**

8

3

10

**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 4. Mảng hiệu**

Cho mảng số nguyên A[] gồm N phần tử, mảng hiệu của mảng a là mảng D[] với D[0] = A[0] và D[i] = A[i] - A[i - 1] . Nhiệm vụ của bạn là xây dựng mảng hiệu của mảng A[]

**Input Format**

* Dòng 1 là N
* Dòng 2 là N số nguyên

**Constraints**

* 1<=N<=10^6
* 1<=A[i]<=10^9

**Output Format**

In ra mảng hiệu của mảng A[]

**Sample Input 0**

5

7 6 9 9 8

**Sample Output 0**

7 -1 3 0 -1

**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 5. Update Operation**

Cho mảng số nguyên A[] gồm N phần tử, có Q thao tác, mỗi thao tác yêu cầu các bạn tăng tất cả các phần tử từ chỉ số L tới chỉ số R lên K đơn vị. Bạn hãy in ra mảng sau khi thực hiện các thao tác trên.

**Input Format**

* Dòng 1 là N và Q
* Dòng 2 là N phần tử trong mảng A[]
* Q dòng tiếp theo mỗi dòng là 3 số L, R, K

**Constraints**

* 1<=N<=10^6
* 1<=Q<=10^5
* 0<=L<=R<=N-1, 0<=A[i],K<=1000

**Output Format**

In ra mảng sau Q thao tác

**Sample Input 0**

7 3

8 5 8 9 7 6 9

0 5 0

0 5 0

1 5 1

**Sample Output 0**

8 6 9 10 8 7 9

**Mảng Cộng Dồn - Mảng Hiệu]. Bài 6. Tổng lớn nhất**

Tèo đang làm các bài toán với truy vấn tổng trên đoạn, mỗi truy vấn yêu cầu các bạn tính tổng các phần tử từ chỉ số L tới chỉ số R, tuy nhiên bài toán này đã quá quen thuộc và dễ dàng có thể dùng mảng cộng dồn để giải quyết vì thế Tèo đưa ra thêm 1 yêu cầu cho bài toán.

Cho trước các truy vấn tổng từ chỉ số L tới chỉ R, bây giờ bạn đã biết trước tất cả Q truy vấn bạn được phép thay đổi thứ tự các phần tử trong mảng 1 lần duy nhất trước khi thực hiện các truy vấn. Hãy thay đổi mảng sao cho tổng các truy vấn trên đoạn đạt được giá trị lớn nhất. Một cách rõ ràng hơn, bạn hãy tính tổng mọi mảng con trong từng truy vấn sau đó cộng lại để đạt được giá trị lớn nhất.

#include <bits/stdc++.h>

using namespace std;

using ll = long long;

int main(){

ios::sync\_with\_stdio(false);

cin.tie(nullptr);

int n, q; cin >> n >> q;

int a[n];

for(int i = 0; i < n; i++){

cin >> a[i];

}

int d[n + 5] = {0};

while(q--){

int l, r; cin >> l >> r;

--l; --r;

d[l] += 1;

d[r + 1] -= 1;

}

int F[n];

F[0] = d[0];

for(int i = 1; i < n; i++){

F[i] = F[i - 1] + d[i];

}

sort(F, F + n, greater<int>());

sort(a, a + n, greater<int>());

ll res = 0;

for(int i = 0; i < n; i++){

res += 1ll \* a[i] \* F[i];

}

cout << res << endl;

}

**Input Format**

* Dòng 1 là N và Q
* Dòng 2 là N số trong mảng A[]
* Q dòng tiếp theo, mỗi dòng là 2 chỉ số L, R của truy vấn

**Constraints**

* 1<=N,Q<=2.10^5
* 1<=A[i]<=2.10^5
* 1<=L<=R<=N

**Output Format**

In ra tổng lớn nhất có thể đạt được.

**Sample Input 0**

6 2

5 2 5 3 5 1

2 5

2 4

**Sample Output 0**

33

**Explanation 0**

Thay đổi mảng A[] thành : {1, 5, 5, 5, 3, 2} khi đó truy vấn từ 2 tới 5 là 18 và truy vấn từ 2 tới 4 là 15 nên tổng lớn nhất là 33.

**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 7. Pha Trà Sữa**

Tèo mới lên đại học và kiếm được một công việc tại hàng trà sữa, có không quá 200000 cốc trà sữa xếp thành 1 hàng ngang, ban đầu mỗi cốc trà sữa có độ ngọt bằng 0. Tèo tiến hành liên tiếp N các thao tác, mỗi thao tác anh ta sẽ thêm 1 gam đường vào các cốc trà sữa từ vị trí L tới vị trí R. Sau cùng những cốc trà sữa có lượng đường >= K gam được chuyển đi để phục vụ khách hàng. Có nhiều câu hỏi, mỗi câu hỏi yêu cầu bạn trả lời trong các cốc trà từ thứ L tới thứ R có bao nhiêu cốc đạt chuẩn để bán cho khách hàng ?

**Input Format**

* Dòng 1 gồm 3 số N, K, Q với Q là số truy vấn
* N dòng tiếp theo mỗi dòng gồm 2 số L, R (thao tác)
* Q dòng tiếp theo mỗi dòng là 2 số L, R (truy vấn)

**Constraints**

* 1<=K<=N<=200000
* 1<=Q<=200000
* 1<=L,R<=200000

**Output Format**

Đối với mỗi truy vấn in ra số lượng cốc trà sữa đạt chuẩn

**Sample Input 0**

5 3 4

2 9

5 8

2 4

2 4

1 6

3 4

5 7

5 8

5 8

**Sample Output 0**

2

2

2

2

**[Mảng Cộng Dồn - Mảng Hiệu]. Bài 8. PrefixPrefix**

Cho mảng A[] có N phần tử và M thao tác, mỗi thao tác yêu cầu bạn tăng các phần tử ở vị trí L tới R của mảng lên D đơn vị (1<=L<=R<=N). Các thao tác này được đánh số từ 1 đến M, ngoài ra Tèo lại muốn thực hiện K truy vấn, mỗi truy vấn Tèo lại thực hiện các thao tác từ x tới y theo số thứ tự thao tác (1<=x<=y<=M). Bạn hãy giúp Tèo in ra mảng A[] sau K truy vấn

**Input Format**

* Dòng 1 là N, M, K
* Dòng 2 là N số trong mảng A[]
* M dòng tiếp theo mỗi dòng gồm 3 số L, R, D
* K dòng tiếp theo mỗi dòng gồm 2 số x, y

**Constraints**

* 1<=N,M,K<=10^5
* 0<=A[i]<=10^5
* 1<=L,R<=N, 0<=D<=10^5
* 1<=x,y<=M

**Output Format**

In ra mảng sau khi thực hiện xong các truy vấn

**Sample Input 0**

6 5 2

1 7 5 2 8 7

2 6 5

1 5 1

1 6 1

2 6 4

2 5 4

2 5

2 4

**Sample Output 0**

5 23 21 18 24 17

**[Two Pointer]. Bài 1. Trộn 2 dãy**

Cho 2 mảng A[] và B[] có N và M phần tử đã được sắp xếp theo thứ tự tăng dần, nhiệm vụ của bạn là trộn 2 mảng này thành một mảng và sắp xếp theo thứ tự tăng dần. Độ phức tạp mong muốn là tuyến tính.

**Input Format**

* Dòng đầu tiên là N và M
* Dòng 2 là N số trong mảng A[]
* Dòng 3 là M số trong mảng B[]

**Constraints**

* 1<=N,M<=10^7
* 1<=A[i],B[i]<= 10^9

**Output Format**

In ra mảng tăng dần sau khi trộn A[] và B[]

**Sample Input 0**

6 6

1 2 4 4 9 10

4 4 5 6 6 8

**Sample Output 0**

1 2 4 4 4 4 5 6 6 8 9 10

**[Two Pointer]. Bài 2. 28Tech\_Smaller**

Cho 2 mảng A[] và B[] có N và M phần tử đã được sắp xếp theo thứ tự tăng dần, nhiệm vụ của bạn là đối với

mỗi phần tử trong mảng B[] hãy đếm xem trong mảng A[] có bao nhiêu phần tử nhỏ hơn nó.

**Input Format**

* Dòng đầu tiên là N và M
* Dòng 2 là N số trong mảng A[]
* Dòng 3 là M số trong mảng B[]

**Constraints**

* 1<=N,M<=10^7
* 1<=A[i],B[i]<= 10^9

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

6 7

2 3 6 6 6 8

5 5 5 6 6 7 10

**Sample Output 0**

2 2 2 2 2 5 6

**[Two Pointer]. Bài 3. 28Tech\_Numer\_Of\_Equal**

Cho 2 mảng A[] và B[] có N và M phần tử đã được sắp xếp theo thứ tự tăng dần, nhiệm vụ của bạn là hãy đếm xem trong 2 mảng tồn tại bao nhiêu cặp i, j sao cho A[i] = B[j]

**Input Format**

* Dòng đầu tiên là N và M
* Dòng 2 là N số trong mảng A[]
* Dòng 3 là M số trong mảng B[]

**Constraints**

* 1<=N,M<=10^7
* 1<=A[i],B[i]<= 10^9

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

6 7

3 3 3 4 7 9

2 3 3 5 6 9 10

**Sample Output 0**

7

**[Two Pointer]. Bài 4. Phát quà Noel**

Nhân dịp lễ Giáng Sinh 2022, 28Tech tổ chức phát quà cho các bạn nhỏ. Có N món quà được xếp thành hàng ngang, mỗi món quà đều có khối lượng cho trước. Tèo là một đứa trẻ cũng như nhiều đứa trẻ khác chỉ muốn có càng nhiều phần quà càng tốt, không cần biết tới khối lượng của từng mốn quà nặng nhẹ ra sao.

Tuy nhiên chiếc túi của Tèo chỉ mảng được trọng lượng tối đa là S. Bạn hãy xác định xem số lượng phần quà mà Tèo có thể lựa chọn tối đa là bao nhiêu để có thể không vượt quá trọng lượng tối đa mà cái túi có thể chịu được. Ngoài ra trong lúc chọn quà Tèo chỉ có thể chọn các phần quà xếp cạnh nhau mà thôi.

**Input Format**

* Dòng đầu tiên là N và S
* Dòng thứ 2 là trọng lượng của N phần quà

**Constraints**

* 1<=N<=10^6
* 1<=S<=10^9
* Trọng lượng các phần quà là số nguyên dương không quá 10^6

**Output Format**

In ra số phần quà tối đa mà Tèo có thể lấy được

**Sample Input 0**

11 10

3 1 4 1 5 3 4 1 5 2 2

**Sample Output 0**

4

**[Two Pointer]. Bài 5. Phát quà Noel 2**

Nhân dịp lễ Giáng Sinh 2022, 28Tech tổ chức phát quà cho các bạn nhỏ. Có N món quà được xếp thành hàng ngang, mỗi món quà đều có khối lượng cho trước. Tí là một đứa trẻ không như nhiều đứa trẻ khác, Tí chỉ muốn chọn ít phần quà càng tốt miễn là tổng các phần quà này lớn hơn hoặc bằng S. Tí chỉ có thể lựa chọn các phần quà đặt cạnh nhau, bạn hãy xác định xem Tí có thể chọn tối thiểu bao nhiêu phần quà để tổng khối lượng của các phần quà lớn hơn hoặc bằng S.

**Input Format**

* Dòng đầu tiên là N và S
* Dòng thứ 2 là trọng lượng của N phần quà

**Constraints**

* 1<=N<=10^6
* 1<=S<=10^9
* Trọng lượng các phần quà là số nguyên dương không quá 10^6

**Output Format**

In ra số lượng phần quà ít nhất có thể hoặc in ra -1 nếu không tồn tại đáp án.

**Sample Input 0**

14 14

4 4 5 1 3 1 3 4 1 1 5 4 1 4

**Sample Output 0**

4

**[Two Pointer]. Bài 6. 28Tech\_SegCount1**

Cho mảng A[] gồm N số nguyên và số nguyên S, nhiệm vụ của bạn là đếm xem có bao nhiêu mảng con các phần tử liên tiếp trong mảng mà tổng không vượt quá S.

**Input Format**

* Dòng đầu tiên là N và S
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=N<=10^6
* 1<=A[i]<=10^6
* 1<=S<=10^9

**Output Format**

In ra số lượng mảng con thỏa mãn

**Sample Input 0**

13 14

5 4 5 2 4 5 1 1 2 5 5 2 2

**Sample Output 0**

47

**[Two Pointer]. Bài 7. 28Tech\_SegCount2**

Cho mảng A[] gồm N số nguyên và số nguyên S, nhiệm vụ của bạn là đếm xem có bao nhiêu mảng con các phần tử liên tiếp trong mảng mà tổng ít nhất bằng S

**Input Format**

* Dòng đầu tiên là N và S
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=N<=10^6
* 1<=A[i]<=10^6
* 1<=S<=10^9

**Output Format**

In ra số lượng mảng con thỏa mãn

**Sample Input 0**

7 20

2 6 4 3 6 8 9

**Sample Output 0**

9

**[Two Pointer]. Bài 8. 28Tech\_SegCount3**

Cho mảng A[] gồm N số nguyên và số nguyên K, nhiệm vụ của bạn là đếm xem có bao nhiêu mảng con các phần tử liên tiếp trong mảng mà số lượng phần từ khác nhau trong mảng con này không vượt quá K.

**Input Format**

* Dòng đầu tiên là N và K
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=K<=N<=10^5
* 1<=A[i]<=10^6

**Output Format**

In ra số lượng mảng con thỏa mãn đề bài

**Sample Input 0**

11 3

5 4 4 5 4 4 2 1 5 2 4

**Sample Output 0**

42

**[Two Pointer]. Bài 9. 28Tech\_SegCount4**

Cho mảng A[] gồm N số nguyên và số nguyên K, nhiệm vụ của bạn là đếm xem có bao nhiêu mảng con các phần tử liên tiếp trong mảng mà độ chênh lệch giữa phần tử lớn nhất và phần tử nhỏ nhất trong mảng con đó không vượt quá K

**Input Format**

* Dòng đầu tiên là N và K
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=N<=10^5
* 1<=A[i],K<=10^6

**Output Format**

In ra số lượng mảng con thỏa mãn đề bài

**Sample Input 0**

10 1

2 3 1 3 1 4 1 3 1 2

**Sample Output 0**

12

**[Two Pointer]. Bài 10. TwoSum**

Cho mảng A[] gồm N phần tử và số nguyên K, hãy kiểm tra xem trong mảng có 2 phần tử A[i], A[j] với i khác j và A[i] + A[j] = K hay không ? Bài này các bạn có thể giải theo 3 cách : Binary search, Hai con trỏ, Map.

**Input Format**

* Dòng đầu tiên là N và K
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=N<=5000
* 1<=A[i],K<=10^9

**Output Format**

In ra YES nếu tồn tại, ngược lại in ra NO

**Sample Input 0**

5 28

2 1 10 5 9

**Sample Output 0**

NO

**Sample Input 1**

7 12

8 3 8 5 5 9 8

**Sample Output 1**

YES

**[Two Pointer]. Bài 11. Sum of three values**

Cho mảng A[] gồm N phần tử và số nguyên K, hãy kiểm tra xem trong mảng có 3 phần tử A[i], A[j], A[k] với i, j, k khác nhau và A[i] + A[j] + A[k] = K hay không ?

**Input Format**

* Dòng đầu tiên là N và K
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=N<=10^5
* 1<=A[i],K<=10^9

**Output Format**

In ra YES nếu tồn tại, ngược lại in ra NO

**Sample Input 0**

7 5

7 2 5 10 10 8 8

**Sample Output 0**

NO

**Sample Input 1**

7 13

1 7 6 3 3 1 8

**Sample Output 1**

YES

**[Two Pointer]. Bài 12. Sum of four values**

Cho mảng A[] gồm N phần tử và số nguyên K, hãy kiểm tra xem trong mảng có 3 phần tử A[i], A[j], A[k], A[l] với i, j, k, l khác nhau và A[i] + A[j] + A[k] + A[l] = K hay không ?

**Input Format**

* Dòng đầu tiên là N và K
* Dòng thứ 2 là N số trong mảng A[]

**Constraints**

* 1<=N<=1000
* 1<=A[i],K<=10^9

**Output Format**

In ra YES nếu tồn tại, ngược lại in ra NO

**Sample Input 0**

6 28

3 6 7 9 1 6

**Sample Output 0**

YES

**[Two Pointer]. Bài 13. Xâu 28**

28tech rất yêu thích con số 28 vì thế anh ta có bài toán sau và nhờ bạn tìm lời giải. Cho một xâu kí tự S chỉ bao gồm kí tự 2 và kí tự 8, bạn hãy tìm xâu con liên tiếp có độ dài lớn nhất mà trong đó chứa nhiều nhất X kí tự 2 và Y kí tự 8.

**Input Format**

* Dòng 1 chứa xâu S
* Dòng 2 chứa 2 số X, Y

**Constraints**

* 2 <=len(S) <= 10^6

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

8228282888

3 3

**Sample Output 0**

6

**[Sliding Window]. Bài 1. Sliding Window**

Cho dãy số A[] gồm có N phần tử và số nguyên dương k. Hãy tìm dãy con liên tiếp có k phần tử sao cho dãy con đó có tổng các phần tử lớn nhất.

**Input Format**

Dòng đầu tiên là số nguyên N và K. Dòng tiếp theo gồm N số nguyên A[i]

**Constraints**

1≤ K ≤ N ≤ 10^7; 1 ≤ A[i] ≤ 10^9;

**Output Format**

Dòng 1 in ra tổng lớn nhất của dãy con K phần tử liền kề. Dòng 2 in ra dãy con đó, nếu có nhiều dãy con có cùng tổng lớn nhất thì in ra dãy con xuất hiện đầu tiên.

**Sample Input 0**

59 4

72 39 5 74 29 83 63 67 69 10 50 86 32 66 96 100 47 32 41 96 9 7 58 97 87 26 21 23 100 52 31 76 43 74 17 72 104 40 99 8 16 27 38 78 69 46 16 58 73 52 49 67 62 42 64 28 46 34 83

**Sample Output 0**

315

72 104 40 99

**[Sliding Window]. Bài 2. Maximum window**

Cho mảng A[] gồm N phần tử và số nguyên K, nhiệm vụ của bạn là in ra số lớn nhất và nhỏ nhất của mọi mảng con cỡ K của mảng A[]

**Input Format**

Dòng đầu tiên là N và K; Dòng thứ 2 gồm N phần tử A[i]

**Constraints**

1<=K<=N<=10^6; 0<=A[i]<=10^9

**Output Format**

In ra số lớn nhất và nhỏ nhất của mọi mảng con trên từng dòng.

**Sample Input 0**

8 2

8 5 8 6 7 6 5 8

**Sample Output 0**

5 8

5 8

6 8

6 7

6 7

5 6

5 8

**[Sliding Window]. Bài 3. Sliding median**

Cho mảng A[] gồm N phần tử và số nguyên K, bạn hãy tìm median của mọi dãy con liên tiếp cỡ K của mảng A[].

**Input Format**

* Dòng 1 là N và K
* Dòng 2 gồm N số trong mảng A[]

**Constraints**

* 1<=K<=N<=2.10^5
* 1<=A[i]<=10^9

**Output Format**

In ra N - K + 1 median của các dãy con

**Sample Input 0**

8 2

7 6 6 2 9 6 2 7

**Sample Output 0**

6 6 2 2 6 2 2

**Sample Input 1**

6 1

7 2 9 2 4 6

**Sample Output 1**

7 2 9 2 4 6

**[Sliding Window]. Bài 4. Smallest window**

Cho xâu S chỉ gồm chữ cái in thường, nhiệm vụ của bạn là tìm chiều dài của xâu con liên tiếp nhỏ nhất có chứa đầy đủ các kí tự khác nhau của S. Ví dụ S = abcaaaabcad thì xâu con bcad có độ dài nhỏ nhất và chứa đầy đủ các kí tự khác nhau của S.

**Input Format**

Dòng duy nhất chứa xâu S

**Constraints**

1<=len(S)<=10^6

**Output Format**

In ra độ dài xâu con nhỏ nhất tìm được

**Sample Input 0**

bcceedeeaedda

**Sample Output 0**

9

**[Sliding Window]. Bài 5. Longest window**

Cho xâu kí tự S, Tèo là một người khá kì quặc, anh ta chỉ thích những thứ không trùng lặp. Vì thế anh ta yêu cầu bạn tìm ra độ dài xâu con dài nhất mà các kí tự trong xâu con đó không có bất cứ một kí tự nào được lặp lại.

**Input Format**

Dòng duy nhất chứa xâu S

**Constraints**

* 1<=len(S)<=10^6
* S chỉ bao gồm chữ cái in thường

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

dabacdadbbdd

**Sample Output 0**

4

**[Sliding Window]. Bài 6. Minimum Window**

Cho 2 xâu kí tự S và T, nhiệm vụ của bạn là tìm xâu con ngắn nhất của S chứa đầy đủ mọi kí tự của xâu T. Ví dụ S = ADOBECODEBANCC, T = CABC thì xâu con BANCC là xâu con ngắn nhất của S chứa đầy đủ mọi kí tự của T

**Input Format**

* Dòng 1 là xâu S
* Dòng 2 là xâu T

**Constraints**

* 1<=len(T),len(S)<=10^6
* S, T chỉ chứa các kí tự in thường

**Output Format**

In ra xâu con nhỏ nhất thỏa mãn và xuất hiện đầu tiên nếu có nhiều xâu thỏa mãn. Nếu không tồn tại đáp án in ra -1

**Sample Input 0**

dbcddbaaccacb

ab

**Sample Output 0**

ba

**Sample Input 1**

ccaadcdcdbdabddb

dbb

**Sample Output 1**

bdab

**[Sliding Window]. Bài 7. Minimum Subarray**

Cho mảng A[] có N phần tử, bạn hãy tìm ra dãy con các phần tử liên tiếp có tổng tối thiểu bằng K. Nếu có nhiều dãy con thỏa mãn hãy in ra dãy con ngắn nhất xuất hiện đầu tiên, trong trường hợp không tồn tại dãy con thỏa mãn in ra -1.

**Input Format**

* Dòng 1 là N và K
* Dòng 2 gồm N số trong mảng A[]

**Constraints**

* 1<=N<=10^6
* 0<=A[i]<=10^6
* 0<=K<=10^9

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

11 7

2 4 0 4 2 3 3 2 0 0 3

**Sample Output 0**

4 0 4

**[Sliding Window]. Bài 8. Duplicate Window**

Cho mảng A[] gồm N phần tử và số nguyên K, bạn hãy xác định xem trong mảng có tồn tại 2 phần tử A[i], A[j] với i khác j sao cho A[i] = A[j] và abs(i - j) <= K hay không. Nếu con in ra YES, ngược lại in ra NO. Bài này các bạn có thể sử dụng sort hoặc sliding window

**Input Format**

* Dòng 1 gồm N và K
* Dòng 2 gồm N số trong mảng A[]

**Constraints**

* 1<=K<=N<=10^6
* 1<=A[i]<=10^6

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

5 5

9 10 7 8 4

**Sample Output 0**

NO

**Sample Input 1**

13 7

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

**Sample Output 1**

YES

**[Sliding Window]. Bài 9. Trồng cây phi lao**

Nhà Tèo ở gần bờ sông, trên bờ sông có trồng N cây phi lao, tuy nhiên có những cây không thể sống được do thời tiết khắc nghiệt của mùa đông miền Bắc. Một ngày nọ Tèo đi dọc bờ sông và tự hỏi, liệu rằng có thể tìm được K cây phi lao con còn sống liên tiếp dọc bờ sông hay không. Có thể sẽ không tồn tại K cây phi lao con còn sống liên tiếp, vì thế Tèo sẽ chuyển những cây phi lao con còn sống ở những ô khác để tạo ra một dãy các cây phi lao sống liên tiếp. Bạn hãy giúp Tèo di chuyển ít nhất cây phi lao nhất có thể, hoặc xác định rằng không thể di chuyển các cây phi lao còn sống lại gần nhau để tạo ra K cây phi lao liên tiếp còn sống.

**Input Format**

* Dòng 1 là N và K
* Dòng 2 mô tả N cây phi lao, 0 tương ứng với cây phi lao chết, 1 tương ứng với cây phi lao sống.

**Constraints**

* 1<=K<=N<=10^6

**Output Format**

In ra số cây phi lao ít nhất cần di chuyển hoặc in ra -1 nếu không thể tạo ra 1 dãy K cây phi lao còn sống.

**Sample Input 0**

11 3

0 0 0 0 1 0 1 0 0 0 1

**Sample Output 0**

1