

BÀI TẬP LẬP TRÌNH

XẾP HÀNG

Lớp trưởng xếp hàng n học sinh trước cửa lớp theo thứ tự chiều cao tăng dần theo cách như sau: di chuyển một học sinh bất kỳ về đầu hàng hoặc cuối hàng cho đến khi được một hàng có chiều cao tăng dần.

Yêu cầu: Hãy giúp lớp trưởng xếp hàng sao cho số lần di chuyển học sinh là ít nhất.

Dữ liệu: Vào từ tập tin văn bản **LINEUP.INP**

- Dòng đầu tiên chứa số nguyên dương n - số học sinh ($1 < n \leq 10^5$)
- Dòng tiếp theo chứa n số nguyên khác nhau đôi một h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$) - chiều cao của n học sinh.

Kết quả: Ghi ra tập tin văn bản **LINEUP.OUT** một số nguyên là số lần di chuyển ít nhất tìm được.

Ví dụ:

LINEUP . INP	LINEUP . OUT
5 3 7 2 6 9	3

HỘP ĐỰNG QUÀ

Bờm có n món quà, món quà thứ i có giá trị c_i . Bờm mua được m chiếc hộp khác nhau, các hộp được đánh số từ 1 đến m . Mỗi chiếc hộp có thể đặt vừa một số món quà nào đó và mỗi món quà chỉ được đặt trong một hộp.

Bờm muốn chọn ra một số món quà, mỗi món quà đặt trong một chiếc hộp để tặng cho cả nhà bạn gái. Bờm cần chọn các món quà để tặng sao cho tổng giá trị của chúng là lớn nhất.

Yêu cầu: Cho danh sách các món quà và thông tin các loại hộp. Tìm tổng giá trị lớn nhất của các món quà được chọn.

Dữ liệu: Vào từ tập tin văn bản **BOXES.INP**

- Dòng đầu tiên chứa 2 số nguyên $n, m (n, m \leq 100)$.
- Mỗi dòng trong n dòng tiếp theo chứa 2 số nguyên c_i, b_i cho biết giá trị của món quà và thứ tự của chiếc hộp có thể chứa được món quà này.

Kết quả: Ghi ra tập tin văn bản **BOXES.OUT** một số nguyên là tổng giá trị lớn nhất của các món quà được chọn.

Ví dụ:

BOXES . INP	BOXES . OUT
5 6 3 1 4 2 1 1 2 3 1 6	10

ĐOẠN 0

Cho dãy số nguyên a_1, a_2, \dots, a_n ($1 \leq n \leq 10^5; |a_i| \leq 10^6$). Hãy tìm một đoạn dài nhất gồm các phần tử liên tiếp trong dãy a_L, a_{L+1}, \dots, a_H có tổng bằng 0

Dữ liệu: Vào từ tập tin văn bản **SZERO.INP**

- Dòng 1: Chứa số n .
- Dòng 2: Chứa n số a_1, a_2, \dots, a_n theo đúng thứ tự cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra tập tin văn bản **SZERO.OUT** hai số L, H tìm được.

Ví dụ:

SZERO . INP	SZERO . OUT
9 2 7 5 -3 -2 4 -9 -2 1	2 8

Dữ liệu vào được cho hợp lý để luôn tồn tại lời giải của bài toán.

XÂY DỰNG MẢNG

Cho dãy n số nguyên a_1, a_2, \dots, a_n . Hãy xây dựng dãy b_1, b_2, \dots, b_n trong đó b_i là chỉ số nhỏ nhất của phần tử đứng sau a_i và lớn hơn a_i . Nói cách khác $b_i = j$ nhỏ nhất ($j > i$) sao cho $a_j > a_i$, $b_i = -1$ nếu không tìm được vị trí j thỏa điều kiện.

Dữ liệu: Vào từ tập tin văn bản **JFS.INP** chứa dãy a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$) không vượt quá 10^6 phần tử, các phần tử cách nhau khoảng trắng.

Kết quả: Ghi ra tập tin văn bản **JFS.OUT** dãy b_1, b_2, \dots, b_n

Ví dụ:

JFS . INP	JFS . OUT
4 7 -8 5 3	2 -1 4 -1 -1

DIỆN TÍCH PHỦ

Cho n hình chữ nhật có cạnh song song với các trục tọa độ và tâm của các hình chữ nhật nằm tại gốc tọa độ. Mỗi hình chữ nhật được xác định bởi chiều rộng X và chiều cao Y .

Yêu cầu: Xác định diện tích vùng mặt phẳng của hệ trục tọa độ bị phủ bởi n hình chữ nhật.

Dữ liệu: Vào từ tập tin văn bản **COVERING.INP**

- Dòng đầu tiên chứa số nguyên dương $n (n \leq 10^6)$.
- Dòng thứ i trong n dòng tiếp theo chứa 2 số nguyên chẵn $X_i, Y_i (2 \leq X, Y \leq 10^7)$ tương ứng chiều rộng và chiều cao của hình chữ nhật thứ i .

Kết quả: Ghi ra tập tin văn bản **COVERING.OUT** một số nguyên duy nhất là diện tích của vùng bị phủ.

Ví dụ:

COVERING . INP	COVERING . OUT
3 8 2 4 4 2 6	28

SỐ NGUYÊN TỐ

Cho hai số nguyên dương n và h , hãy liệt kê tất cả các số nguyên tố trong phạm vi từ 1 tới n có tổng các chữ số bằng h .

Dữ liệu: Vào từ tập tin văn bản **HPRIME.INP** dòng chứa hai số nguyên dương $n, h (n \leq 10^6, h \leq 10^9)$

Kết quả: Ghi ra tập tin văn bản **HPRIME.OUT**

- Dòng 1 ghi số k là số những số nguyên tố trong phạm vi từ 1 tới n có tổng các chữ số bằng h
- k dòng tiếp theo, mỗi dòng ghi một số nguyên tố thỏa mãn yêu cầu đề ra, các số nguyên tố phải được liệt kê theo thứ tự tăng dần.

Ví dụ:

HPRIME . INP	HPRIME . OUT
500 16	7 79 97 277 349 367 439 457

NGHIỆM NGUYÊN TỐ

Xét phương trình có dạng $x + y + z = K$ (K là một số nguyên dương). Bộ (x, y, z) được gọi là nghiệm nguyên tố của phương trình nếu $x + y + z = K$, $x \leq y \leq z$ và x, y, z đều là số nguyên tố.

Phương trình này có thể có vô số nghiệm, tuy nhiên, với mỗi K cho trước thì số nghiệm nguyên tố của phương trình hoàn toàn có thể xác định được.

Dữ liệu: Vào từ tập tin văn bản **EQUATION.INP** gồm nhiều dòng, mỗi dòng chứa một số nguyên dương K ($K < 5000$) là một trường hợp dữ liệu cần kiểm tra.

Kết quả: Ghi ra tập tin văn bản **EQUATION.OUT** gồm nhiều dòng, mỗi dòng ứng với dữ liệu của một trường hợp kiểm tra của dữ liệu vào là số bộ nghiệm nguyên tố của phương trình tương ứng.

Ví dụ:

EQUATION . INP	EQUATION . OUT
5	0
8	1

ĐỊNH ĐỀ BERTRAND

Định đề Bertrand phát biểu: với một số tự nhiên $n > 0$ luôn tồn tại một số nguyên tố p mà $n < p \leq 2n$

Yêu cầu: Cho n , kiểm tra định đề Bertrand bằng cách đếm số lượng số nguyên tố nằm trong $[n + 1, 2n]$

Dữ liệu: Vào từ tập tin văn bản **DDB.INP**

- Dòng đầu tiên chứa T là số bộ dữ liệu.
- T dòng sau, mỗi dòng tương ứng với một bộ dữ liệu là số nguyên $n \leq 10^6$.

Kết quả: Ghi ra tập tin văn bản **DDB.OUT** trên nhiều dòng, mỗi dòng ghi số lượng số nguyên tố trong đoạn tương ứng dữ liệu vào

Ví dụ:

DDB . INP	DDB . OUT
2	1
1	1
3	

SỐ SIÊU NGUYÊN TỐ

Xét dãy vô hạn P chứa các số nguyên tố sắp xếp theo thứ tự tăng dần, Các số nguyên tố được đánh số từ 1 trở đi. Như vậy ta có $P_1 = 2, P_2 = 3, P_3 = 5, \dots, P_{52} = 239, \dots$

Số nguyên tố P_i được gọi là siêu nguyên tố nếu i cũng là một số nguyên tố. Như vậy, 3 và 5 là các số siêu nguyên tố, còn 239 – không phải là siêu nguyên tố. Các số siêu nguyên tố được sắp xếp theo thứ tự tăng dần và đánh số từ 1 trở đi.

Yêu cầu: Cho số nguyên $k (1 \leq k \leq 500)$. Hãy xác định số siêu nguyên tố thứ k .

Dữ liệu: Vào từ tập tin văn bản **SUPPRIME.INP** gồm một dòng chứa số nguyên k .

Kết quả: Ghi ra tập tin văn bản **SUPPRIME.OUT** số siêu nguyên tố tìm được.

Ví dụ:

SUPPRIME . INP	SUPPRIME . OUT
3	11

SỐ NGUYÊN TỐ ĐỐI XỨNG

Các số nguyên tố liệt kê theo thứ tự tăng dần 2, 3, 5, 7, 11, 13, ... tạo thành một dãy và đánh số bắt đầu từ 1. Gọi p_i là số nguyên tố thứ i , ta nói p_i là số nguyên tố đối xứng nếu nó bằng trung bình cộng của 2 số nguyên tố liền trước và liền sau nó. Nói cách khác p_i là số nguyên tố đối xứng nếu thỏa điều kiện:

$$p_i = \frac{p_{i-1} + p_{i+1}}{2}$$

Như vậy 10 số nguyên tố đối xứng đầu tiên là: 5, 53, 157, 173, 211, 257, 263, 373, 563, 593.

Yêu cầu: Cho số nguyên n . Cho biết n có phải là số nguyên tố đối xứng hay không.

Dữ liệu: Vào từ tập tin văn bản **SYMPRIME.INP**

- Dòng đầu tiên chứa số nguyên $t (1 \leq t \leq 10^5)$ – số lượng số n cần kiểm tra.
- Mỗi dòng trong t dòng tiếp theo chứa số nguyên $n (1 \leq n \leq 2 \times 10^7)$.

Kết quả: Ghi ra tập tin văn bản **SYMPRIME.OUT** gồm t dòng, mỗi dòng ghi thông báo YES hoặc NO là câu trả lời tương ứng với câu hỏi cho trong dữ liệu vào.

Ví dụ:

SYMPRIME . INP	SYMPRIME . OUT
3	NO
11	YES
5	YES
373	

ĐOẠN CON

Cho dãy n số nguyên a_1, a_2, \dots, a_n và số nguyên dương k .

Yêu cầu: Tìm đoạn con liên tiếp gồm $\geq k$ số nguyên trong dãy, sao cho tổng các số nguyên thuộc đoạn là lớn nhất.

Dữ liệu: Vào từ tập tin văn bản **SUBSEQ.INP**

- Dòng 1 chứa hai số nguyên dương n, k ($1 \leq k \leq n \leq 10^6$)
- n dòng tiếp theo, dòng thứ i chứa một số nguyên a_i . ($|a_i| \leq 1000$)

Kết quả: Ghi tập tin văn bản **SUBSEQ.OUT** một số nguyên là tổng các giá trị đoạn con tìm được theo yêu cầu.

Ví dụ:

SUBSEQ . INP	SUBSEQ . OUT
8 3 -20 90 -30 -20 80 -70 -60 125	120

TỔNG ĐOẠN CON

Cho dãy số nguyên không âm: a_1, a_2, \dots, a_n . Xếp các số trong dãy theo thứ tự trên một vòng tròn theo chiều kim đồng hồ. Ta gọi một đoạn con của vòng tròn là một dãy các số liên tục theo chiều kim đồng hồ trên vòng tròn đó.

Yêu cầu: Hãy tìm một đoạn con có tổng bằng số nguyên S cho trước sao cho độ dài của đoạn con đó là ngắn nhất.

Dữ liệu: Vào từ tập tin văn bản **SUMSEQ.INP**

- Dòng đầu chứa hai số nguyên dương $n (n \leq 10^5)$ và S
- Dòng tiếp theo chứa dãy gồm n số nguyên không âm: $a_1, a_2, \dots, a_n (a_i \leq 10^9)$.

Kết quả: Ghi ra tập tin văn bản **SUMSEQ.OUT** hai số nguyên i và l , trong đó i là vị trí bắt đầu của đoạn con, l là độ dài của đoạn con. Nếu không tìm được dãy con nào, ghi ra một số 0 duy nhất.

Ví dụ:

SUMSEQ . INP	SUMSEQ . OUT
5 3 0 1 0 0 2	5 3

HAI TẬP

Cho tập gồm n số nguyên $\{1, 2, \dots, n\}$. Hãy chia tập thành 2 tập con có tổng bằng nhau.

Dữ liệu: Vào từ tập tin văn bản **TWOSETS.INP** chứa số nguyên $n (1 \leq n \leq 10^6)$.

Kết quả: Ghi ra tập tin văn bản **TWOSETS.OUT**

- Dòng đầu tiên ghi YES tương ứng với có cách chia, ngược lại in NO.
- Nếu có cách chia thì in ra một cách chia trên 4 dòng như sau:
 - + Dòng đầu tiên ghi số nguyên a – số phần tử của tập thứ nhất.
 - + Dòng thứ hai ghi a số nguyên là các phần tử của tập thứ nhất.
 - + Dòng thứ ba ghi số nguyên b – số phần tử của tập thứ hai.
 - + Dòng thứ tư ghi b số nguyên là các phần tử của tập thứ hai.

Ví dụ:

TWOSETS . INP	TWOSETS . OUT
7	YES 4 1 2 4 7 3 3 5 6
6	NO

CHUỖI HỢP LỆ

Một chuỗi được gọi là hợp lệ nếu tất cả kí tự trong chuỗi có tần số xuất hiện bằng nhau. Một chuỗi nếu có thể xóa đúng 1 kí tự tại đúng 1 vị trí nào đó để chuỗi trở thành hợp lệ thì cũng được gọi là chuỗi hợp lệ.

Chẳng hạn chuỗi $s = \text{"abc"}$ là chuỗi hợp lệ, chuỗi $s = \text{"acbc"}$ cũng là chuỗi hợp lệ vì ta có thể xóa 1 kí tự c để trở thành $s = \text{"abc"}$ hoặc $s = \text{"acb"}$ đều là chuỗi hợp lệ. Chuỗi $s = \text{"abccc"}$ không phải là chuỗi hợp lệ.

Yêu cầu: Cho chuỗi s độ dài không quá 10^5 kí tự latin in thường. Cho biết s có phải là chuỗi hợp lệ hay không.

Dữ liệu: Vào từ tập tin văn bản **VALIDSTR.INP** chứa chuỗi s .

Kết quả: Ghi ra tập tin văn bản **VALIDSTR.OUT** thông báo YES hoặc NO cho biết câu trả lời tương ứng.

Ví dụ:

VALIDSTR.INP	VALIDSTR.OUT
abcdefghijklghgfedecba	YES
aabbcd	NO

TỪ DÀI NHẤT

Cho xâu ký tự S chỉ gồm các chữ cái in hoa và dấu cách. Một dãy liên tiếp các chữ cái in hoa trong xâu S được gọi là một từ.

Yêu cầu: Hãy cho biết có bao nhiêu từ trong xâu ký tự và chiều dài của từ dài nhất.

Dữ liệu: Vào từ tập tin văn bản **WORDS.INP** gồm một dòng chứa xâu ký tự S gồm không quá 10^6 ký tự.

Kết quả: Ghi ra tập tin văn bản **WORDS.OUT** hai số nguyên là số từ trong xâu và độ dài của từ dài nhất.

Ví dụ:

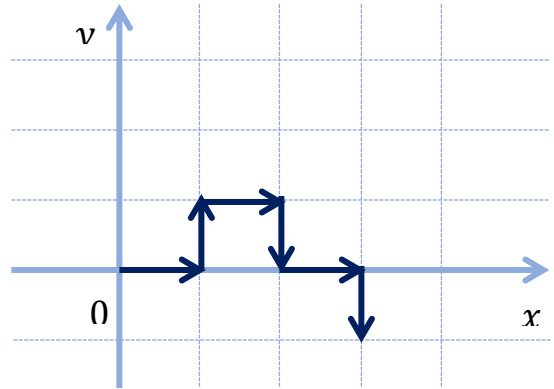
WORDS . INP	WORDS . OUT
TOP OF THE WORLD	4 5

ROBOT DI CHUYỂN

Cho lưới nguyên Oxy. Điểm nguyên (x_1, y_1) và điểm nguyên (x_2, y_2) được gọi là kề nhau nếu thỏa điều kiện $|x_1 - x_2| + |y_1 - y_2| = 1$.

Một robot ban đầu đứng tại gốc tọa độ. Ở mỗi bước, robot sẽ di chuyển sang một điểm nguyên kề với vị trí hiện tại.

Từ bước di chuyển thứ 2 trở đi, robot có thể đi tiếp theo hướng cũ, rẽ sang trái, rẽ sang phải hay trở lại vị trí trước đó.



Trong ví dụ ở hình bên, từ ô $(0, 0)$, robot đi đến $(1, 0)$, rẽ trái sang ô $(1, 1)$, rẽ phải sang ô $(2, 1)$, rẽ phải sang ô $(2, 0)$, rẽ trái sang ô $(3, 0)$ cuối cùng rẽ phải sang ô $(3, -1)$.

Yêu cầu: cho tọa độ các điểm nguyên mà robot đã đi qua. Hãy đếm xem robot đã rẽ phải bao nhiêu lần.

Dữ liệu: Vào từ tập tin văn bản **ROBOT.INP** có cấu trúc sau:

- Dòng đầu tiên chứa 1 số nguyên dương n ($2 \leq n \leq 10000$) là tổng số điểm nguyên mà robot đã đi qua (kể cả vị trí xuất phát là gốc tọa độ)
- Dòng thứ i trong n dòng tiếp theo chứa 2 số nguyên x_i, y_i là tọa độ điểm nguyên mà robot đã đi qua.

Kết quả: Ghi ra tập tin văn bản **ROBOT.OUT** gồm một số nguyên là số lần robot đã rẽ phải

Ví dụ:

ROBOT . INP	ROBOT . OUT
7 0 0 1 0 1 1 2 1 2 0 3 0 3 -1	3

ĐƯỜNG ĐI TRÊN BẢNG

Cho bảng kích thước $n \times m$, mỗi ô của bảng chứa một số nguyên có giá trị từ 0 đến 9. Một đường đi trên bảng được gọi là hợp lệ nếu thỏa:

- Đi theo chiều ngang từ trái qua phải
- Đi theo chiều dọc từ trên xuống
- Đi theo đường chéo song song với đường chéo chính từ trên xuống.
- Độ dài đường đi không vượt quá 6 ô.
- Các chữ số trên đường đi tạo thành một số nguyên dương gọi là chi phí đường đi và chi phí đường đi là số nguyên tố.

Yêu cầu: Hãy tìm tất cả đường đi hợp lệ và tính tổng chi phí của tất cả đường đi hợp lệ.

Dữ liệu: Vào từ tập tin văn bản **TPATH.INP**

- Dòng đầu chứa 2 số nguyên $n, m (1 \leq n, m \leq 1500)$
- Mỗi dòng trong n dòng tiếp theo chứa dãy gồm m số nguyên (có giá trị từ 0 đến 9).

Kết quả: Ghi ra tập tin văn bản **TPATH.OUT** 2 số nguyên: số thứ nhất là số lượng đường đi hợp lệ – số thứ hai là tổng chi phí của tất cả đường đi hợp lệ.

Ví dụ:

TPATH . INP	TPATH . OUT
4 5 1 2 3 4 5 6 7 8 9 0 0 9 8 7 6 5 4 3 2 1	19 1547

Giải thích: có 19 đường đi hợp lệ: 2, 2, 3, 3, 5, 5, 7, 7, 17, 23, 43, 61, 67, 71, 79, 83, 89, 97, 883.

THU GOM RÁC

Để cải thiện môi trường nhằm thu hút khách du lịch người ta quyết định tiến hành thu gom và tiêu hủy rác dọc bờ biển. Toàn bộ bờ biển thuộc khu du lịch được chia thành n đoạn đánh số từ 1 đến n . Khảo sát sơ bộ cho thấy ở đoạn thứ i có x_i tấn rác.

Xe liên hợp kiểu mới thu gom, phân loại và chế biến rác được đưa ra vận hành thử nghiệm. Trong một khoảng thời gian hoạt động liên tục xe có thể thu gom và chế biến không quá t tấn rác. Vì là lần vận hành thử nghiệm nên các kỹ sư chế tạo rất thận trọng, muốn chọn một khúc bờ biển nào đó gồm một số đoạn liên tiếp để tiện theo dõi và đánh giá.

Yêu cầu: Hãy xác định có bao nhiêu cách chọn khác nhau nếu chỉ dựa vào tiêu chí đảm bảo sao cho xe không phải xử lý quá t tấn rác. Mỗi đoạn trong khúc đã chọn phải được làm sạch, tức là thu gom hết rác trong đoạn đó. Hai khúc gọi là khác nhau nếu tồn tại một đoạn có ở trong khúc này và không có trong khúc kia.

Dữ liệu: Vào từ tập tin văn bản **TRASH.INP**

- Dòng đầu tiên chứa 2 số nguyên n và t ($1 \leq n \leq 10^6; 1 \leq t \leq 10^9$).
- Dòng thứ hai chứa n số nguyên x_1, x_2, \dots, x_n ($1 \leq x_i \leq 10^6$). Tổng các x_i không vượt quá 10^9 .

Kết quả: Ghi ra tập tin văn bản **TRASH.OUT** một số nguyên – số cách lựa chọn khác nhau có thể thực hiện.

Ví dụ:

TRASH . INP	TRASH . OUT
9 10 11 1 2 1 1 5 10 2 3	19

Giải thích: có 19 đoạn khác nhau: $[1], [2], [1], [1], [5], [10], [2], [3], [1\ 2], [2\ 1], [1\ 1], [1\ 5], [2\ 3], [1\ 2\ 1], [2\ 1\ 1], [1\ 1\ 5], [1\ 2\ 1\ 1], [2\ 1\ 1\ 5], [1\ 2\ 1\ 1\ 5]$.

LÙA BÒ

Bò được Phú Ông giao chăn đàn bò gồm n con được đánh số từ 1 đến n . Trong lúc Bò đang mê ngủ, các con bò đã lên vào vườn trồng hoa quý của Phú Ông. Ngay khi Bò phát hiện thì các con bò bắt đầu ăn hoa và Bò quyết định phải đưa tất cả chúng về chuồng ngay lập tức.

Con bò thứ i có tốc độ ăn d_i cây hoa trong mỗi phút chờ đợi và nó đang ở cách chuồng một quãng đường mất t_i phút để di chuyển. Ở mỗi thời điểm, Bò chỉ có thể đưa một con bò về chuồng. Bò mất $2 \times t_i$ phút để đưa con bò thứ i về chuồng và quay trở lại.

Bò đang lo lắng vì không biết đưa cả đàn bò về chuồng thế nào để tổng số hoa quý của Phú Ông bị thiệt hại là ít nhất.

Yêu cầu: Cho 2 dãy số nguyên t_i và d_i tương ứng với thời gian di chuyển về chuồng và tốc độ ăn hoa của con bò thứ i . Hãy tính số cây hoa bị thiệt hại ít nhất sau khi cả đàn bò về chuồng.

Dữ liệu: Vào từ tập tin văn bản **COWS.INP**

- Dòng đầu tiên chứa số nguyên n ($2 \leq n \leq 10^5$)
- Dòng thứ i trong n dòng tiếp theo chứa 2 số nguyên t_i, d_i ($1 \leq t_i \leq 2 \times 10^6$; $1 \leq d_i \leq 100$).

Kết quả: Ghi ra tập tin văn bản **COWS.OUT** một số nguyên là số cây hoa ít nhất bị đàn bò ăn.

Ví dụ:

COWS . INP	COWS . OUT
6 3 1 2 5 2 3 3 2 4 1 1 6	86

CHUỖI HẠT

Mai mua được một chuỗi hạt đá rất đẹp, mỗi viên đá mang 1 trong 26 màu riêng biệt và được kí hiệu bằng các chữ cái a,b,c,...,z. Vì chuỗi hạt khá dài nên Mai muốn cắt ra một đoạn để làm vòng đeo tay hoặc vòng cổ. Theo Mai, chiếc vòng có tính thẩm mỹ chỉ khi nó có đủ k màu khác nhau.

Yêu cầu: Cho chuỗi hạt được mô tả bằng xâu kí tự in thường. Hãy giúp Mai cắt ra một đoạn hạt ngắn nhất sao cho có đủ k màu khác nhau.

Dữ liệu: Vào từ tập tin văn bản **BEADS.INP**

- Dòng đầu tiên chứa chuỗi kí tự chữ cái in thường có độ dài không quá 10^5 .
- Dòng tiếp theo chứa số nguyên $k(k \geq 1)$.

Kết quả: Ghi ra tập tin văn bản **BEADS.OUT** một số nguyên là độ dài nhỏ nhất của đoạn hạt cắt được hoặc -1 nếu không tìm được cách cắt thỏa yêu cầu.

Ví dụ:

BEADS . INP	BEADS . OUT
ddaahhceeeaaabb 4	5