**Bài 1: Mã bài sort**

Mirko là một chuyên gia giải mã. Anh ta ăn cắp một thông điệp. Thông điệp gồm n số nhỏ hơn hoặc bằng C. Mirko tin rằng việc sắp xếp các số này lại theo thứ tự là được thông điệp ban đầu . Quy tắc sắp xếp như sau

Chuỗi số được sắp xếp sao cho với hai số X, Y. X xuất hiện trước Y nếu số lần xuất hiện của X trong chuỗi lớn hơn Y. Nếu số lần xuất hiện bằng nhau. Thì số nào xuất hiện trước được sắp trước.

Input: sort.inp

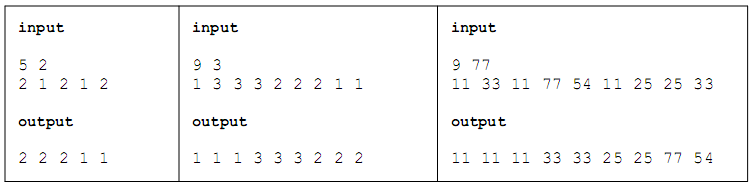
Dòng đầu tiên chứa hai số nguyên N (1 ≤ N ≤ 1 000) là độ dài của dãy số, C (1 ≤ C ≤ 1 000 000 000).

dòng tiếp theo chứa số n số nguyên nhỏ hơn C

OUTPUT : sort.inp

Chuỗi số đã sắp xếp

Ví dụ



Bài 2: Mã bài **THIEF**

Có một ngôi làng có n ngôi nhà được đánh số từ 1 đến n. Các nhà này đều được bắt điện thoại. TOLENT rất tò mò và đặt máy nghe trộm. Máy đặt ở vị trí i có thể nghe được tất cả các cuộc gọi từ hai phía ( trước i và sau i). Sau một tháng TOLENT đã nghe trộm được khá nhiều cuộc gọi. Hãy viết chương trình cho biết ít nhất là bao nhiêu cuộc gọi được thực hiện.

+ Dữ liệu vào: File văn bản **THIEF.INP**

Dòng đầu tiên chứa hai số nguyên N (số máy theo dõi) (1 ≤ N ≤ 100 000), M số ngôi nhà (N<M ≤ 1000000000).

N dòng tiếp theo mỗi dòng chứa hai số nguyên dương Pi (1 ≤ Pi < M), và Ci (1 ≤ Ci ≤ 1000 000000) là vị trí đặt máy nghe trộm và tổng số cuộc gọi máy i nghe được. Máy i đặt ở giữa ngôi nhà i và i+1.

+ Dữ liệu ra: Ghi vào file văn bản **THIEF.OUT**

Số cuộc gọi ít nhất

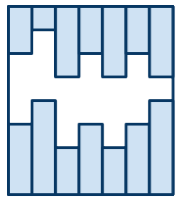
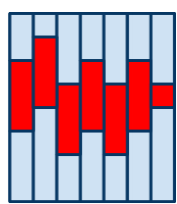
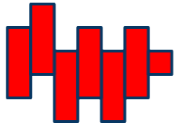
Ví dụ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **THIEF.INP** | **THIEF.OUT** | **THIEF.INP** | **THIEF.OUT** | **THIEF.INP** | **THIEF.OUT** |
| 3 4  3 1  2 2  1 1 | 2 | 2 3  1 23  2 17 | 23 | 3 9  7 2  8 3  3 4 | 5 |

**Bài 3**

Một tòa tháp cổ được khóa bởi các ổ khóa rất phức tạp. Mỗi ổ khóa dài L cm và rộng W cm và gồm 3 phần: Cạnh trên, cạnh dưới và khoảng trống chính giữa. Các cạnh phía trên và các cạnh phía dưới, mỗi cạnh được mô tả bởi một chuỗi gồm L số nguyên không âm r1 r2 r3 ... rL biểu diễn độ rộng của cạnh tại vị trí đó. Chìa khóa của mỗi ổ khóa là một hình sao cho bỏ vừa khít vào khoảng trống của ổ khóa.

Ví dụ hình ảnh một ổ khóa dài 7cm và rộng 8cm với chìa khóa tương ứng là:

1. Ổ khóa b) Ổ khóa có chìa khóa c) Chìa hhóa

Các cạnh trên của ổ khóa được mô tả bởi một chuỗi số là: 2, 1, 3, 2, 3, 2, 3

Các cạnh dưới của ổ khóa được mô tả bởi một chuỗi số là: 3, 4, 2, 3, 2, 3, 4

Nhưng sau khi để ý người ta có thể dùng chìa khóa của ổ này mở cho ổ khác. Để tiết kiệm chi phí người ta tìm cách khắc ít chìa khóa nhất. Em hay lập trình tìm số ít nhất chìa khóa cần sản xuất để mở được tất cả các ổ khóa

INPUT file : kletva.inp

Dòng đầu tiên chứa đựng 3 số nguyên dương W (1 ≤ W ≤ 108), độ rộng của tất cả các ổ khóa, L (1 ≤ L ≤ 1000), độ dài tất cả các ổ khóa, N (1 ≤ N ≤ 100), số ổ khóa khác nhau.

2N dòng tiếp mô tả tất cả các ổ khóa. Hai dòng liên tiếp nhau mô tả một ổ khóa. Dòng đầu mô tả cạnh trên, dòng sau mô tả cạnh dưới. Giữa hai cạnh có ít nhất một khoảng trống là 1cm

OUTPUT file: kletva.out