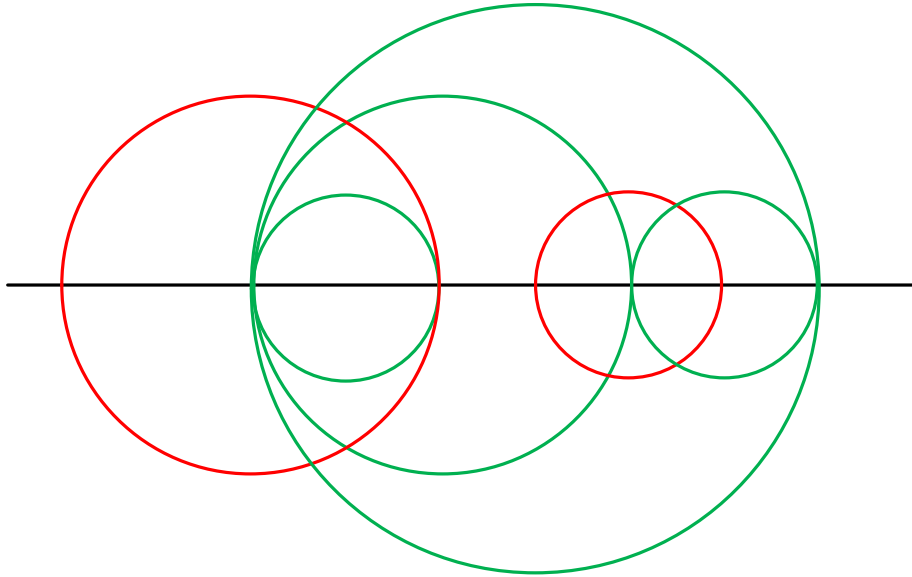


## A. Vòng tròn không cắt [NCUTCIR]

Cho  $N$  đường tròn có tâm nằm trên trục  $Ox$ , đường tròn thứ  $i$  có hoành độ tâm  $c_i$  và bán kính  $r_i$ . Các đường tròn cùng bán kính đều có tâm khác nhau.



Hãy xác định số lượng ít nhất các đường tròn cần loại bỏ để trong số các đường tròn còn lại không có cặp đường tròn nào cắt nhau (nhưng có thể tiếp xúc với nhau).

**Dữ liệu (ncutcir.inp)**

- Dòng 1: số nguyên  $N$  ( $1 \leq N \leq 1000$ )
- Dòng 2...  $N + 1$ : dòng  $i$  chứa hai số nguyên  $c_i, r_i$  ( $1 \leq c_i, r_i \leq 100$ ).

**Kết quả (ncutcir.out)**

- Dòng 1: số nguyên kết quả.

**Ví dụ**

ncutcir.inp	ncutcir.out
6 2 1 5 1 6 1 1 2 3 2 4 3	2

## B. Điểm trên biên [PRBORDER]

Trên mặt phẳng tọa độ cho  $n$  điểm nguyên (lattice point) và  $m$  hình chữ nhật có các cạnh cùng phương với các trục  $Ox, Oy$ , các đỉnh có tọa độ nguyên.

Hãy xác định số điểm trong  $n$  điểm đã cho nằm trên biên của mỗi hình chữ nhật.

### Dữ liệu (prborder.inp)

- Dòng 1: số nguyên  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ )
- Dòng 2...  $n + 1$ : dòng  $i + 1$  chứa 2 số nguyên  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ) là tọa độ điểm thứ  $i$
- Dòng  $n + 2$ : số nguyên  $m$  ( $1 \leq m \leq 10^5$ )
- Dòng  $n + 3 \dots n + 2 + m$ : mỗi dòng chứa 4 số nguyên  $x_1, y_1, x_2, y_2$  ( $1 \leq x_1 < x_2 \leq 10^9; 1 \leq y_1 < y_2 \leq 10^9$ ) là tọa độ đỉnh dưới-trái  $(x_1; y_1)$  và đỉnh trên-phải  $(x_2; y_2)$  của một hình chữ nhật.

### Kết quả (prborder.out)

- Dòng 1...  $m$ : dòng  $i$  ghi số nguyên đáp án cho hình chữ nhật  $i$ .

### Ví dụ

prborder.inp	prborder.out
6	3
1 2	4
3 2	0
2 3	1
2 5	
4 4	
6 3	
4	
2 2 4 4	
2 2 6 5	
3 3 5 6	
5 1 6 6	

## C. Người bên cạnh [KPSWAP]

Có  $n$  người (đánh số  $1, 2, \dots, n$ ) đứng thành vòng tròn theo chiều ngược chiều kim đồng hồ mặt hướng vào tâm vòng tròn. Vị trí ban đầu của người số 1 gọi là vị trí xuất phát. Họ thực hiện  $k$  lượt di chuyển như sau: ở lượt  $i$ , người đang ở vị trí xuất phát sẽ đổi chỗ với người bên phải mình  $p_i$  lần ( $p_i$  là số nguyên tố thứ  $i$ ).

Hãy xác định người đứng bên cạnh người số hiệu  $t$  khi kết thúc  $k$  lượt di chuyển.

### Dữ liệu (kpswap.inp)

- Dòng 1: ba số nguyên  $n, k, t$  ( $3 \leq n \leq 5 \times 10^6, 1 \leq k \leq 5 \times 10^5, 1 \leq t \leq n$ )

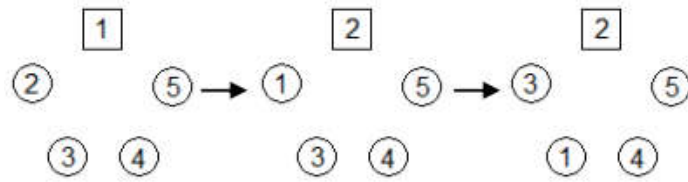
### Kết quả (kpswap.out)

- Dòng 1: hai số nguyên là số hiệu người bên phải và người bên trái của người  $t$ .

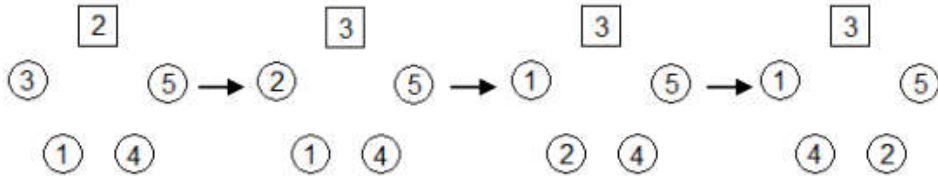
### Ví dụ

kpswap.inp	kpswap.out
5 3 1	3 5

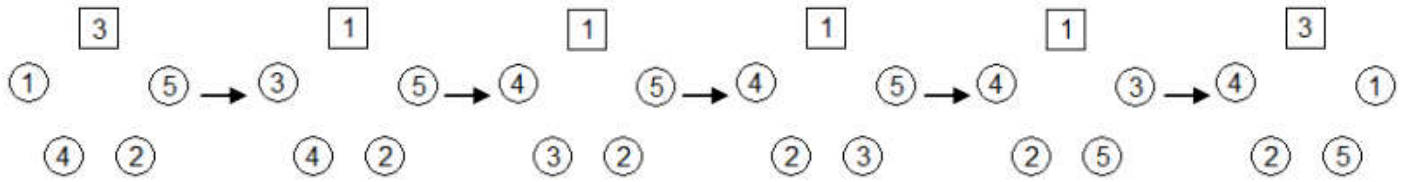
Lượt 1: người 1 đổi chỗ 2 lần



Lượt 2: người 2 đổi chỗ 3 lần



Lượt 3: người 3 đổi chỗ 5 lần



## D. Bán kẹo [CANDSTOR]

Bờm là chủ một cửa hàng bánh kẹo. Kẹo được đóng trong  $N$  túi, mỗi túi có một có lượng kẹo nhất định. Khi có khách hàng tới hỏi mua  $K$  viên kẹo, Bờm sẽ mang ra một số túi có tổng đúng  $K$  viên. Nếu điều này không thể làm được, khách hàng sẽ bỏ đi.

Chính vì vậy, Bờm muốn biết cách đóng gói hiện tại có thể phục vụ bao nhiêu yêu cầu khác nhau của khách hàng và muốn tìm cách đóng gói để đáp ứng nhiều yêu cầu nhất. Cụ thể Bờm muốn mở một túi có số lượng  $P$  và cho thêm kẹo vào đó để có số lượng  $Q$ .

Chẳng hạn, nếu cửa hàng có 4 túi với số lượng tương ứng là  $(1, 3, 4, 4)$  thì Bờm có thể đáp ứng được 9 yêu cầu khác nhau của khách:  $1, 3, 4, 5, 7, 8, 9, 11, 12$ . Nếu Bờm mở gói có số lượng 4 và bổ sung kẹo vào đó để có số lượng là 9 thì tập gói kẹo sẽ là  $(1, 3, 4, 9)$  và đáp ứng được 13 yêu cầu khác nhau là  $1, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14, 16, 17$ .

Cho  $N$  và  $b_1, b_2, \dots, b_N$  trong đó  $b_i$  là số lượng kẹo trong túi  $i$ . Hãy xác định  $P$  và  $Q$  để số yêu cầu có thể đáp ứng được là lớn nhất. Nếu có nhiều cách lựa chọn  $P$  thì đưa ra  $P$  nhỏ nhất. Với giá trị  $P$  này, nếu có nhiều cách chọn  $Q$  thì đưa ra  $Q$  nhỏ nhất.

**Dữ liệu (candstor.inp)**

- Dòng 1: số nguyên  $N$  ( $2 \leq N \leq 100$ )
- Dòng 2:  $N$  số nguyên  $b_1, b_2, \dots, b_N$  ( $1 \leq b_i \leq 7000$ )

**Kết quả (candstor.out)**

- Dòng 1: hai số nguyên  $P$  và  $Q$  tìm được.

**Ví dụ**

candstor.inp	candstor.out
4 1 3 4 4	4 9

## E. Thứ tự hoán vị [ORDERPSW]

Xét các hoán vị của  $K$  số nguyên  $1, 2, \dots, K$ , theo quan hệ thứ tự từ điển, mỗi hoán vị có một số thứ tự xác định trong phạm vi  $[1, 2, \dots, K!]$ .

Cho  $P = (p_1, p_2, \dots, p_K)$  là một hoán vị của  $K$  số nguyên  $1, 2, \dots, K$ . Ta kí hiệu  $P(u, v)$  là hoán vị nhận được từ  $P$  bằng cách đổi chỗ hai phần tử  $p_u, p_v$ . Hãy xác định số thứ tự của  $P(u, v)$ .

### Dữ liệu (orderpsw.inp)

- Dòng 1: hai số nguyên  $K, N$  ( $2 \leq K \leq 3 \times 10^5; 1 \leq N \leq 10^5$ )
- Dòng 2:  $K$  số nguyên  $p_1, p_2, \dots, p_K$
- Dòng 3 ...  $N + 2$ : dòng  $i + 2$  chứa hai số nguyên  $u_i, v_i$

### Kết quả (orderpsw.out)

- Dòng 1 ...  $N$ : số nguyên là số thứ tự của  $P(u_i, v_i)$  theo module  $(10^9 + 7)$ .

### Ví dụ

orderpsw.inp	orderpsw.out
5 3 1 5 4 2 3 1 3 2 3 2 5	91 17 9

## F. Tăng lương [SALINC]

Bờm lập một công ty, ban đầu chỉ có một thành viên là Bờm, có số hiệu 1, nhận mức lương  $s_1$ . Trong quá trình phát triển, công ty lần lượt nhận thêm các thành viên mới số hiệu  $2, 3, \dots, n$ . Nhân viên số hiệu  $i$  vào công ty sẽ nhận một trong các nhân viên  $1, 2, \dots, i - 1$  là lãnh đạo trực tiếp. Mức lương đàm phán được của nhân viên  $i$  là  $s_i$ .

Mỗi khi một thành viên công ty được hưởng mức lương cao hơn lãnh đạo trực tiếp, lương của người lãnh đạo đó sẽ được tăng lên mức bằng nhân viên đó.

Cho biết quá trình nhận thêm thành viên của công ty, hãy xác định số người được tăng lương mỗi khi công ty thêm người mới.

### Dữ liệu (salinc.inp)

- Dòng 1: số nguyên  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ )
- Dòng 2: số nguyên  $s_1$
- Dòng 3 ...  $n + 1$ : dòng  $i + 1$  ghi hai số nguyên  $s_i, b_i$  là lương khởi điểm của nhân viên  $i$  và số hiệu lãnh đạo trực tiếp của nhân viên  $i$ .  
( $1 \leq s_i \leq 10^9 \forall i = 1 \div n$ )

### Kết quả (salinc.out)

- Dòng 1 ...  $n$ : dòng  $i$  ghi số nguyên là số lượt tăng lương trong công ty xảy ra sau khi nhận thêm nhân viên  $i$ .

Ví dụ

Input	Output
7	0
5000	1
4500 0	0
6000 0	2
4000 1	4
5500 3	1
7000 4	0
6300 2	
6300 2	

## G. Đồn thùng [BINPCKLR]

Cho dãy  $N$  thùng rỗng đánh số  $1, 2, \dots, N$  từ trái qua phải, thùng thứ  $i$  có kích thước  $b_i \leq M$ . Nhằm tạo chỗ trống, ta có thể thực hiện một số lần thao tác đồn thùng như sau: chọn hai thùng  $i, j$  thỏa mãn  $i < j; b_i < b_j$  đặt thùng  $i$  vào trong thùng  $j$ , vị trí  $i$  trở thành vị trí trống.

Yêu cầu đối với dãy thùng sau khi đồn

- Mỗi thùng chỉ được chứa tối đa một thùng rỗng khác
- Các chỗ trống tạo được phải là một đoạn liên tiếp từ bên trái của dãy thùng ban đầu
- Các thùng không rỗng (chứa một thùng rỗng) phải là một đoạn liên tiếp từ bên trái của dãy thùng sau khi đồn.

Hãy xác định số chỗ trống  $K$  lớn nhất tạo được.

Dữ liệu (binpcklr.inp)

- Dòng 1: hai số nguyên  $M, N$  ( $1 \leq M \leq 1000; 1 \leq N \leq 20000$ )
- Dòng 2:  $N$  số nguyên  $b_1, b_2, \dots, b_N$

Kết quả (binpcklr.out)

- Dòng 1: số nguyên  $K$

Ví dụ

binpcklr.inp	binpcklr.out
5 10	4
2 2 1 4 3 2 5 4 2 3	