

## d5rec

Cho  $n$  hình chữ nhật, hình thứ  $i$  ( $1 \leq i \leq n$ ) có kích thước  $a_i \times b_i$ . Hình chữ nhật kích thước  $a \times b$  có thể lồng vào hình chữ nhật kích thước  $a' \times b'$  nếu  $(a \leq a' \text{ và } b \leq b')$  hoặc  $(a \leq b' \text{ và } b \leq a')$ .

**Yêu cầu:** Chọn ra nhiều hình nhất để lần lượt xếp lồng vào nhau, mỗi hình chứa trực tiếp một hình khác.

### Input

- Dòng đầu chứa số nguyên  $n$  ( $n \leq 10^5$ );
- Dòng thứ  $i$  ( $1 \leq i \leq n$ ) trong  $n$  dòng sau chứa hai số  $a_i, b_i$  ( $1 \leq a_i, b_i \leq 10^9$ ).

### Output

- Dòng đầu chứa số nguyên  $k$  là số hình chọn được;
- Dòng thứ hai chứa  $k$  số là chỉ số các hình được chọn, các số theo thứ tự các hình từ nhỏ đến lớn.

Input	Output
4	3
2 3	2 1 4
1 2	
1 4	
3 3	

## Dãy chữ số

An và Bình cùng chơi trò chơi với dãy số như sau: An viết liên tiếp một dãy gồm  $n$  chữ số thập phân, tiếp theo Bình tách dãy chữ số trên thành các nhóm chữ số để nhận được một dãy số. Sau đó cả hai bạn cùng tiến hành tìm dãy con tăng dài nhất từ dãy số mới nhận được.

Ví dụ: An viết dãy chữ số thập phân 314159265358979, nếu Bình tách dãy trên thành dãy số gồm 6 số: 3, 14, 159, 26, 53, 58979 thì cả hai bạn sẽ tìm được dãy con tăng dài nhất gồm 5 số là: 3, 14, 26, 53, 58979.

Nhưng nếu Bình tách thành dãy số gồm 10 số: 3, 1, 4, 1, 5, 9, 26, 53, 58, 979 thì cả hai bạn sẽ tìm được dãy con tăng dài nhất gồm 8 số là: 3, 4, 5, 9, 26, 53, 58, 979.

**Yêu cầu:** Cho dãy chữ số thập phân mà An viết, hỏi với cách chơi như trên thì hai bạn có thể tìm được dãy con tăng dài nhất tối đa là bao nhiêu phần tử?

## Input

- Dòng đầu tiên ghi số nguyên dương  $n$  ( $1 \leq n \leq 1000$ ).
- Dòng thứ hai là một xâu gồm  $n$  chữ số thập phân.

## Output

- Một số duy nhất là độ dài của dãy con tăng dài nhất tìm được.

Input	Output	Input	Output
15 314159265358979	8	10 1233456789	9

## Vectơ

Cho  $n$  vectơ, mỗi vectơ có  $m$  thành phần. Hai vectơ  $U(u_1, u_2, \dots, u_m)$  và vectơ  $V(v_1, v_2, \dots, v_m)$  được gọi là có “quan hệ” với nhau nếu tồn tại  $(\alpha_1, \alpha_2, \dots, \alpha_m)$  là hoán vị của  $m$  thành phần vectơ  $U$  và  $(\beta_1, \beta_2, \dots, \beta_m)$  là hoán vị của  $m$  thành phần vectơ  $V$  sao cho  $\alpha_i \leq \beta_i$  ( $\forall i = 1..m$ ) hoặc  $\alpha_i \geq \beta_i$  ( $\forall i = 1..m$ )

**Yêu cầu:** Chia  $n$  vectơ cho trước thành ít nhóm nhất sao cho trong mỗi nhóm không có hai vectơ nào có “quan hệ” với nhau.

*Giới hạn của bài toán:*  $n \leq 1000, m \leq 20$

## Input

- Dòng đầu ghi hai số  $n, m$ ;
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi  $m$  thành phần của vectơ thứ  $i$ .

## Output

- Dòng đầu ghi  $k$  là số nhóm ít nhất được chia.
- $k$  dòng tiếp theo, dòng thứ  $i$  dòng ghi:  $s$  là số vector trong nhóm; tiếp theo là  $s$  chỉ số của các vectơ được chọn vào nhóm thứ  $i$ .

Input	Output
4 3	3
1 1 1	1 1
1 2 3	2 2 3
2 2 2	1 4
2 1 1	

## Tập cực trị

Xét tập  $A$  gồm  $n$  xâu, mỗi xâu độ dài  $m$  và chỉ gồm các ký tự “0”, “1” hoặc “\*”. Với một xâu, ta có thể biến đổi để nhận được xâu nhị phân bằng cách thay thế mỗi ký tự “\*” trong xâu thành ký tự “0” hoặc “1”. Biến đổi  $n$  xâu của tập  $A$  để nhận được tập xâu nhị phân  $B$  có lực lượng nhỏ nhất. Biến đổi  $n$  xâu của tập  $A$  để nhận được tập xâu nhị phân  $C$  có lực lượng lớn nhất.

**Yêu cầu:** Cho  $n$  xâu, tìm cách biến đổi  $n$  xâu để nhận được tập xâu nhị phân  $B$  có lực lượng nhỏ nhất và tập xâu nhị phân  $C$  có lực lượng lớn nhất.

## Input

- Dòng đầu chứa hai số nguyên dương  $n, m$  ( $m \leq 100$ );
- $n$  dòng sau, mỗi dòng một xâu độ dài  $m$  chỉ gồm các ký tự “0”, “1” hoặc “\*” mô tả  $n$  xâu của tập  $A$ .

## Output

- Dòng đầu là số nguyên là lực lượng của tập  $B$ ; tiếp theo là  $n$  xâu nhị phân mô tả cách biến đổi từng xâu để nhận được tập  $B$ , các đại lượng cách nhau đúng một dấu cách;
- Dòng thứ hai là số nguyên là lực lượng của tập  $C$ ; tiếp theo là  $n$  xâu nhị phân mô tả cách biến đổi từng xâu để nhận được tập  $C$ , các đại lượng cách nhau đúng một dấu cách.

Input	Output
4 3	2 001 010 001 010
***	4 000 010 001 111
01*	
001	
*1*	

## Cách tính điểm:

Có 20 test, mỗi test 5.0 điểm. Gọi  $s_B, s_C$  tương ứng là lực lượng của tập  $B, C$  do thí sinh tìm được và kết quả của Ban giám khảo tương ứng là  $w_B, w_C$ , khi đó số điểm bạn đạt được cho mỗi test là  $2.5 \times \min \left\{ 1, \left( \frac{w_B}{s_B} \right)^5 \right\} + 2.5 \times \min \left\{ 1, \left( \frac{s_C}{w_C} \right)^5 \right\}$ .

**Subtask 1 (50%) :**  $n \leq 10$ ;

**Subtask 2 (50%):**  $n \leq 1000$ .

## Sắp xếp hoán vị (cyclesz)

Alice và Bob chọn một số nguyên dương  $n$  ( $n \leq 1000$ ), sau đó Bob bí mật tạo ra một hoán vị của  $0, 1, \dots, n-1$ . Alice được phép yêu cầu Bob đổi chỗ hai vị trí  $x, y$  ( $0 \leq x < y \leq n-1$ ) của hoán vị, sau khi đổi xong Bob sẽ cho Alice biết số chu kì trong hoán vị mới.

Một chu kì của hoán vị  $P$  là một dãy các chỉ số  $(i_1, i_2, \dots, i_k)$  sao cho:  $P(i_1) = i_2; P(i_2) = i_3; \dots; P(i_{k-1}) = i_k; P(i_k) = i_1$ .

Bạn cần xây dựng hàm `void sortPermutation(int n)` để nhận được dãy sắp xếp tăng dần với lần ít lần gọi hàm `int performSwap(int x, int y)`.

Mỗi lần gọi hàm `int performSwap(int x, int y)`, hệ thống thực hiện trao đổi hai vị trí và trả lại chu kì trong hoán vị mới.

Chương trình cần `#include "cycles.h"`

## Cài đặt

```
#include <bits/stdc++.h>
#include "cycles.h"

// Xây dựng hàm
void sortPermutation(int n) {
    ...
}

// Hàm sortPermutation có thể gọi performSwap(int x, int y)
```

## Chấm điểm

Số $s$ , số lần gọi hàm <code>performSwap</code>	Điểm
$s > 10^7$	0%
$s > 10^6$	10%
$s > 9 \times 10^4$	20%
$s > 3 \times 10^4$	60%
$s \leq 3 \times 10^4$	100%