

SQSORT

Có $2n + 1$ cấu trúc dữ liệu, trong đó $n + 1$ hàng đợi (Q) và n ngăn xếp (S) xếp xen kẽ nhau từ trên xuống, các cấu trúc dữ liệu được đánh số từ 1 đến $2n + 1$. Cụ thể, các hàng đợi được đánh số $1, 3, \dots, 2n + 1$, các ngăn xếp được đánh số $2, 4, \dots, 2n$. Ban đầu, có k số nguyên trên cấu trúc dữ liệu 1 (hàng đợi có số hiệu 1). Mỗi lượt bạn được phép thực hiện hành động *pop* với cấu trúc dữ liệu i rồi *push* giá trị đó vào cấu trúc dữ liệu $i + 1$.

Yêu cầu: Hãy chỉ ra một dãy các hành động để sắp xếp tất cả k số theo thứ tự tăng dần và các số này nằm trên cấu trúc dữ liệu thứ $2n + 1$ (hàng đợi có số hiệu $2n + 1$).

Input

- Dòng đầu chứa hai số nguyên dương n, k ($k \leq 2^n; n \leq 20$);
- Dòng thứ hai gồm k số nguyên, các số có giá trị tuyệt đối không vượt quá 10^6 ;

Output

- Dòng đầu chứa số nguyên s là số thao tác cần thực hiện;
- Dòng thứ hai gồm s số nguyên mô tả dãy thao tác.

Dữ liệu vào	Kết quả ra
2 3	12
1 2 2	1 1 1 2 2 2 3 3 3 4 4 4

rsort

Xét dãy gồm n số là một hoán vị của $1, 2, \dots, n$ và thao tác sau: Chọn một tập các số, rút ra khỏi dãy, các số lấy ra theo thứ tự từ đầu đến cuối dãy, sau đó lần lượt lấy từng số đặt vào đầu dãy.

Ví dụ, với dãy gồm 5 số 5, 4, 3, 2, 1, nếu chọn tập các số 4, 3, 1 khi đó dãy mới là 1, 3, 4, 5, 2.

Yêu cầu: Cho hai dãy $a_1, a_2, \dots, a_n$ và $b_1, b_2, \dots, b_n$ đều là hoán vị của $1, 2, \dots, n$. Hãy tìm dãy thao tác để biến đổi dãy thứ nhất thành dãy thứ hai với số thao tác không vượt quá $\lceil \log_2 n \rceil$.

Input

- Dòng đầu chứa số nguyên dương n ;
- Dòng thứ hai gồm n số nguyên dương $a_1, a_2, \dots, a_n$;
- Dòng thứ ba gồm n số nguyên dương $b_1, b_2, \dots, b_n$.

Output

- Dòng đầu chứa số nguyên r là số thao tác cần thực hiện;
- Tiếp theo là r dòng, mỗi dòng có khuôn dạng:
 - o Số đầu tiên là số s là số phần tử được chọn;
 - o Tiếp theo là s số được chọn.

Input	Output
5	2
5 4 3 2 1	3 4 3 1
2 5 1 3 4	2 5 2

Subtask 1: $n \leq 5$;

Subtask 2: $n \leq 10$;

Subtask 3: $n \leq 10^4$;

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%

Chọn điểm

Các nhà khoa học đang nghiên cứu để xây dựng trạm vũ trụ quốc tế theo tiêu chuẩn mới. Ban đầu, họ sẽ lắp đặt nó ở trung tâm hàng không vũ trụ trên mặt đất sau đó dùng tàu vũ trụ để đưa trạm lên quỹ đạo. Họ biểu diễn trung tâm bằng hệ tọa độ đề-các hai chiều, trên đó đánh dấu sẵn 29 điểm (được đánh số từ 0 đến 28) có tọa độ là các số nguyên không âm, và là các vị trí thuận lợi cho việc vận chuyển vật liệu. Tiếp theo, họ cần chọn ra 8 điểm trong số đó để lắp đặt 8 mô-đun của trạm. Để thuận tiện cho việc đưa trạm vũ trụ vào quỹ đạo sau này, 8 điểm được chọn phải có trọng tâm là tọa độ nguyên. Nghĩa là, trung bình cộng hoành độ các điểm này là một số nguyên và trung bình cộng tung độ các điểm này cũng là một số nguyên.

Do sơ suất nên phương án chọn 8 điểm đã bị mất. Là một kỹ sư tài năng, Bob được giao nhiệm vụ thiết kế và vận hành một máy tính đơn giản để tìm 8 điểm mà trọng tâm của chúng có tọa độ nguyên. Máy có 64 ô nhớ được đánh số từ 0 đến 63, mỗi ô nhớ đều lưu trữ được các số nguyên không âm có 30 bit, giá trị ô nhớ thứ i được kí hiệu là A_i . Các phép toán mà máy có thể thực thi là:

- **add(i, j, k):** Tính $A_i + A_j$ và gán kết quả vào A_k , nếu kết quả là một số nguyên có nhiều hơn 30 bit thì sẽ chỉ được giữ lại 30 bit cuối.
- **sub(i, j, k):** Tính $A_i - A_j$ và gán kết quả vào A_k , nếu kết quả là một số âm thì sẽ được đặt bằng 0.
- **half(i, j):** Tính phần nguyên của $A_i / 2$ và gán kết quả vào A_j .

Lưu ý, các chỉ số i, j, k có thể bằng nhau và $0 \leq i, j, k \leq 63$.

Ngoài ra, máy có một đèn tín hiệu với rất nhiều dây dẫn đầu vào, mỗi dây có thể được cắm vào một ô nhớ. Đèn sẽ sáng khi trong số các ô nhớ được cắm vào, tồn tại hai ô có giá trị bằng nhau.

Hiện tại, tọa độ của 29 điểm đã được nạp vào máy, hoành độ của điểm thứ i được nạp vào ô nhớ $2i$ và tung độ của điểm thứ i được nạp vào ô nhớ $2i + 1$. Các ô nhớ còn lại đều có giá trị bằng 0.

Yêu cầu

Hãy giúp Bob đưa ra các lệnh tính toán, thay đổi các dây dẫn của đèn và quan sát trạng thái đèn để tìm ra được chỉ số của 8 điểm mà ban đầu trọng tâm của chúng có tọa độ nguyên, sao cho số lần quan sát đèn là càng ít càng tốt và số lệnh tính toán sử dụng là không quá 10^5 . Nếu có nhiều cách chọn ra 8 điểm, chỉ cần đưa ra một cách bất kỳ trong số đó.

Tương tác

Đây là bài toán tương tác không thích ứng. Hệ thống cung cấp thư viện **cpointlib.h** chứa các hàm sau:

- **void add(int i, int j, int k):** Thực hiện phép toán add.
- **void sub(int i, int j, int k):** Thực hiện phép toán sub.
- **void half(int i, int j):** Thực hiện phép toán half.
- **bool ask(string S):** Cắm dây dẫn vào các ô nhớ và quan sát đèn. S là xâu nhị phân độ dài 64, kí tự thứ i là '1' nếu ô nhớ i được cắm vào, ngược lại là '0'. Hàm trả về true nếu đèn sáng, false nếu không sáng.

Chi tiết cài đặt

Thí sinh cần cài đặt hàm sau trong file **cpoint.cpp**:

```
#include "cpointlib.h"
vector<int> solve() {
    // Code của thí sinh
}
```

Hàm solve phải trả về một vector chứa 8 chỉ số của các điểm được chọn.

Chấm điểm

Gọi Q là số lần quan sát đèn trong phương án của thí sinh. Thí sinh sẽ nhận được:

- 0 điểm nếu $Q > 276$;
- $(0.2 \times 0.95^{(Q-174)}) \times 100\%$ số điểm nếu $174 \leq Q \leq 276$;
- $(0.2 + 0.2 \times 0.95^{(Q-94)}) \times 100\%$ số điểm nếu $94 \leq Q < 174$;
- $(0.4 + 0.5 \times 0.8^{(Q-76)}) \times 100\%$ số điểm nếu $76 \leq Q < 94$;
- $(0.9 + 0.1 \times 0.8^{(Q-66)}) \times 100\%$ số điểm nếu $66 < Q < 76$;
- 100% số điểm nếu $Q \leq 66$.

Ràng buộc

Các hoành độ và tung độ của 29 điểm đều là số nguyên không âm và nhỏ hơn 2^{30} .

Subtask

- **Subtask 1 (20 điểm):** Có ít nhất 8 điểm trong 29 điểm mà cả tung độ và hoành độ đều chia hết cho 8.
- **Subtask 2 (80 điểm):** Không có ràng buộc nào thêm.