

## Permutation

Các Pharaoh sử dụng chuyển động tương đối và lực hấp dẫn của các hành tinh để tăng tốc phi thuyền của họ. Giả sử một phi thuyền sẽ đi qua  $n$  hành tinh với tốc độ quỹ đạo lần lượt là  $p[0], p[1], \dots, p[n-1]$ . Đối với mỗi hành tinh, các nhà khoa học của các Pharaoh có thể chọn tăng tốc phi thuyền bằng cách sử dụng hành tinh này hoặc không. Để tiết kiệm năng lượng, sau khi tăng tốc bởi một hành tinh có tốc độ quỹ đạo  $p[i]$ , phi thuyền không thể được tăng tốc bằng cách sử dụng bất kỳ hành tinh nào có tốc độ quỹ đạo  $p[j] < p[i]$ . Nói cách khác, các hành tinh được chọn tạo thành một **dãy con tăng dần** của  $p[0], p[1], \dots, p[n-1]$ . Một dãy con của  $p$  là một dãy nhận được từ  $p$  bằng cách giữ nguyên hoặc xóa một số phần tử của  $p$ . Ví dụ:  $[0]$ ,  $[], [0, 2]$  và  $[0, 1, 2]$  là các dãy con của  $[0, 1, 2]$ , nhưng  $[2, 1]$  thì không phải.

Các nhà khoa học đã xác định rằng có tổng cộng  $k$  cách khác nhau chọn một nhóm hành tinh để tăng tốc phi thuyền nhưng họ đã mất dữ liệu của tất cả các tốc độ quỹ đạo (ngay cả giá trị của  $n$ ). Tuy nhiên, họ nhớ rằng  $(p[0], p[1], \dots, p[n-1])$  là một hoán vị của  $0, 1, \dots, n-1$ . Hoán vị là một dãy mà mỗi phần tử là số nguyên từ  $0$  đến  $n-1$ , mỗi số xuất hiện đúng một lần. Nhiệm vụ của bạn là tìm một hoán vị  $p[0], p[1], \dots, p[n-1]$  có thể xảy ra với độ dài đủ nhỏ.

Bạn cần giải bài toán đối với  $q$  phi thuyền khác nhau. Với mỗi phi thuyền  $i$ , bạn nhận được một số nguyên  $k_i$ , đại diện cho số cách khác nhau chọn một nhóm hành tinh để tăng tốc phi thuyền. Nhiệm vụ của bạn là tìm một dãy các tốc độ quỹ đạo có độ dài  $n_i$  đủ nhỏ sao cho có đúng  $k_i$  cách chọn một dãy hành tinh có tốc độ quỹ đạo tăng dần.

## Chi tiết cài đặt

Bạn cần cài đặt hàm sau:

```
int[] construct_permutation(int64 k)
```

- $k$ : là số lượng dãy con tăng dần mong muốn.
- Hàm này cần trả về một mảng gồm  $n$  phần tử trong đó mỗi phần tử nằm trong đoạn từ  $0$  đến  $n-1$  tính cả hai đầu mút.
- Mảng được trả về phải là một hoán vị hợp lệ có chính xác  $k$  dãy con tăng dần.
- Hàm này được gọi tối đa  $q$  lần. Mỗi lần gọi cần được xem là một trường hợp riêng biệt.

## Ràng buộc

- $1 \leq q \leq 100$
- $2 \leq k_i \leq 10^{18}$  (với mọi  $0 \leq i \leq q - 1$ )

## Subtask

1. (10 điểm)  $2 \leq k_i \leq 90$  (với mọi  $0 \leq i \leq q - 1$ ). Nếu tất cả các hoán vị bạn sử dụng có độ dài tối đa là 90 và đúng, bạn nhận được 10 điểm, ngược lại 0 điểm.
2. (90 điểm) Không có giới hạn gì thêm. Với subtask này, gọi  $m$  là độ dài của hoán vị dài nhất mà bạn đã sử dụng trong tất cả các trường hợp. Khi đó, điểm của bạn được tính theo bảng sau:

Điều kiện	Điểm
$m \leq 90$	90
$90 < m \leq 120$	$90 - \frac{(m-90)}{3}$
$120 < m \leq 5000$	$80 - \frac{(m-120)}{65}$
$m > 5000$	0

## Ví dụ

### Ví dụ 1

Xét lời gọi sau:

```
construct_permutation(3)
```

Hàm này cần trả về một hoán vị với chính xác 3 dãy con tăng dần. Một kết quả có thể là  $[1, 0]$ , có  $[]$  (dãy con trống),  $[0]$  và  $[1]$  là các dãy con tăng dần.

### Ví dụ 2

Xét lời gọi sau:

```
construct_permutation(8)
```

Hàm này cần trả về một hoán vị với chính xác 8 dãy con tăng dần. Một kết quả có thể là  $[0, 1, 2]$ .

## Trình chấm mẫu

Trình chấm mẫu đọc đầu vào theo khuôn dạng sau:

- dòng 1:  $q$
- dòng  $2 + i$  ( $0 \leq i \leq q - 1$ ):  $k_i$

Với mỗi  $k_i$ , trình chấm mẫu ghi ra trên một dòng duy nhất giá trị trả về của hàm `construct_permutation`, hoặc một thông báo lỗi nếu có.