

## პერმუტაცია

ფარაონები პლანეტების ფარდობით მოძრაობასა და გრავიტაციას იყენებენ თავიანთი კოსმოსური ხომალდების დასაჩქარებლად დაუშვან, კოსმოსური ხომალდი გაივლის  $n$  პლანეტას ორბიტალური სიჩქარეებით  $p[0], p[1], \dots, p[n-1]$  თანმიმდევრობით. თითოეული პლანეტისთვის ფარაონის მეცნიერებს შეუძლიათ აირჩიონ დააჩქარონ კოსმოსური ხომალდი ამ პლანეტის გამოყენებით თუ არა. ენერგიის დაზოგვის მიზნით,  $p[i]$ , ორბიტალური სიჩქარის მქონე პლანეტის მიერ აჩქარების შემდეგ, კოსმოსური ხომალდი არ შეიძლება აჩქარდეს უფრო ნაკლები ორბიტალური სიჩქარის მქონე პლანეტის გამოყენებით  $p[j] < p[i]$ . სხვა სიტყვებით, არჩეულმა პლანეტებმა უნდა შექმნან  $p[0], p[1], \dots, p[n-1]$ -ის **ზრდადი ქვემიმდევრობა**.  $p$ -ს ქვემიმდევრობა ეწოდება მიმდევრობას, რომელიც მიიღება  $p$ -დან მასში ნული ან მეტი ელემენტის წაშლით. მაგალითად,  $[0], [], [0, 2]$ , და  $[0, 1, 2]$  წარმოადგენენ  $[0, 1, 2]$ -ის ქვემიმდევრობებს, ხოლო  $[2, 1]$  - არა.

მეცნიერებმა დაადგინეს, რომ არსებობს პლანეტების არჩევის სულ  $k$  განსხვავებული ვარიანტი კოსმოსური ხომალდის დასაჩქარებლად, მაგრამ მათ დაკარგეს ჩანაწერი ორბიტალური სიჩქარეების შესახებ (მათ შორის  $n$ -ის მნიშვნელობაც). თუმცა მათ ახსოვთ, რომ  $(p[0], p[1], \dots, p[n-1])$  არის პერმუტაცია  $0, 1, \dots, n-1$ . პერმუტაცია წარმოადგენს მიმდევრობას, რომელიც შეიცავს ყველა მთელ რიცხვს  $0$ -დან  $(n-1)$ -მდე ზუსტად ერთხელ. თქვენი ამოცანაა აღადგინოთ მიმდევრობა  $p[0], p[1], \dots, p[n-1]$  საკმარისად მცირე სიგრძით.

თქვენ დაგჭირდებათ პრობლემის გადაჭრა  $q$  სხვადასხვა კოსმოსური ხომალდისთვის.  $i$ -ური კოსმოსური ხომალდისთვის, თქვენ მიიღებთ მთელ რიცხვს  $k_i$ , რომელიც წარმოადგენს პლანეტების ნაკრების არჩევის სხვადასხვა გზების რაოდენობას კოსმოსური ხომალდის დასაჩქარებლად. თქვენი ამოცანაა იპოვოთ ორბიტალური სიჩქარეების თანმიმდევრობა საკმარისად მცირე  $n_i$  სიგრძით ისე, რომ შეიძლებოდეს ზუსტად  $k_i$  განსხვავებული გზის არჩევა პლანეტათა ორბიტალური სიჩქარეების მზარდი თანმიმდევრობით.

## იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

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

- $k$ : ზრდადი ქვემიმდევრობების სასურველი რაოდენობა.
- ამ პროცედურამ უნდა დააბრუნოს  $n$  ელემენტისაგან შედგენილი მასივი, რომლის ელემენტთა მნიშვნელობები მოთავსებულია  $0$ -დან  $(n-1)$ -მდე ჩათვლით.

- დაბრუნებული მასივი უნდა იყოს სწორი პერმუტაცია, რომელსაც შეიცავს ზუსტად  $k$  ცალ ზრდად ქვემიმდევრობას.
- ეს პროცედურა სულ გამოიძახება  $q$ -ჯერ. ყოველი გამოძახება ხდება განსხვავებული სცენარისთვის.

## შეზღუდვები

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

## ქვეამოცანები

1. (10 ქულა)  $2 \leq k_i \leq 90$  (ყველა  $0 \leq i \leq q - 1$ ). თუკი თქვენს მიერ გამოყენებული ყველა პერმუტაცია სიგრძით 90-ს არ აღემატება და ისინი სწორია, მაშინ მიიღებათ 10 ქილას, წინააღმდეგ შემთხვევაში - 0-ს.
2. (90 ქულა) დამატებითი შეზღუდვების გარეშე. ამ ქვეამოცანაში  $m$ -ს ექნება მაქსიმალური მნიშვნელობა (გადანაცვლების მაქსიმალური სიგრძე), ხოლო თქვენი შეფასება გამოითვლება შემდეგი ცხრილის მიხედვით:

პირობა	შეფასება
$m \leq 90$	90
$90 < m \leq 120$	$90 - \frac{(m-90)}{3}$
$120 < m \leq 5000$	$80 - \frac{(m-120)}{65}$
$m > 5000$	0

## მაგალითი

### მაგალითი 1

განვიხილოთ შემდეგი გამოძახება:

```
construct_permutation(3)
```

ამ პროცედურამ უნდა დააბრუნოს გადანაცვლება ზუსტად 3 ზრდადი ქვემიმდევრობით. შესაძლო პასუხია  $[1, 0]$ , რომელიც შეიცავს  $[]$  (ცარიელი ქვემიმდევრობა),  $[0]$  და  $[1]$  ზრდად ქვემიმდევრობებს.

### მაგალითი 2

განვიხილოთ შემდეგი გამოძახება:

```
construct_permutation(8)
```

ამ პროცედურამ უნდა დააბრუნოს გადანაცვლება ზუსტად 8 ზრდადი ქვემიმდევრობით. შესაძლო პასუხია  $[0, 1, 2]$ .

## სანიმუშო გრაფერი

სანიმუშო გრაფერი კითხულობს შესატამ მონაცემებს შემდეგი ფორმატით:

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

სანიმუშო გრაფერი ბეჭდავს ერთადერთ სტრიქონს, რომელიც ყოველი  $k_i$ -თვის შეიცავს `construct_permutation`-ის მიერ დაბრუნებულ მნიშვნელობებს, ან ბეჭდავს შეტყობინებას შეცდომის შესახებ.