

順列 (Permutation)

ファラオは惑星の相対的な運動と惑星の重力を利用して宇宙船を加速させる。宇宙船が n 個の惑星を通過する。 n 個の惑星の軌道速度は通過する順に $p[0], p[1], \dots, p[n-1]$ である。ここで、 $p[0], p[1], \dots, p[n-1]$ は $0, 1, \dots, n-1$ の並べ替えである。それぞれの惑星ごとに、ファラオの科学者はその惑星を利用して宇宙船を加速させるかどうかを選ぶことができる。エネルギーを節約するために、軌道速度 $p[i]$ の惑星を利用して加速させた後は、軌道速度 $p[j]$ ($p[j] < p[i]$) の惑星を利用して加速させることができない。すなわち、宇宙船を加速させる惑星の軌道速度を順に並べたとき、 $p[0], p[1], \dots, p[n-1]$ の**増加部分列**となっていなければならない。

科学者は、宇宙船を加速させる惑星の選び方が全部で k 通りあることを特定したが、惑星の軌道速度のデータを失ってしまった。あなたの課題は、軌道速度の列 $p[0], p[1], \dots, p[n-1]$ を復元することである。

あなたは q 個の異なる宇宙船について問題を解かなければならない。各宇宙船 i について、あなたは宇宙船を加速させる惑星の選び方の数を表す整数 k_i を受け取る。あなたの課題は、軌道速度の列 $p[0], p[1], \dots, p[n_i-1]$ であって、軌道速度が単調増加となる惑星の部分列の選び方がちょうど k_i 通りとなるもののうち、長さ n_i ができるだけ小さいものを見つけることである。

実装の詳細

以下の関数を実装せよ:

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

- k : 増加部分列の選び方がちょうど k 通りであることを表す。
- この関数が返す配列の長さを n として、配列の各要素は 0 以上 $n-1$ 以下でなければならない。
- 返される配列はちょうど k 個の増加部分列を持つ順列でなければならない。
- この関数は q 回呼び出される。それぞれの呼び出しは別々のシナリオとして扱われる。

制約

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

小課題

1. (10 点) $2 \leq k_i \leq 90$ ($0 \leq i \leq q-1$)
2. (90 点) 追加の制約はない。全シナリオにおけるあなたが返した順列の長さの最大値を m として、この小課題の得点は以下のように与えられる:

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

入出力例

入出力例 1

以下の関数呼び出しを考える:

```
construct_permutation(3)
```

この関数はちょうど 3 個の増加部分列を持つ順列を返さなければならない。[1, 0] はこの一例であり、次の 3 個の増加部分列を持つ: [] (空の部分列), [0], [1]

入出力例 2

以下の関数呼び出しを考える:

```
construct_permutation(8)
```

この関数はちょうど 8 個の増加部分列を持つ順列を返さなければならない。[0, 1, 2] はこの一例である。

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を読み込む:

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

採点プログラムのサンプルは、各 k_i ごとに `construct_permutation` の返り値を含む 1 行を出力する。エラーが起きた場合は、その代わりにエラーメッセージを出力する。