Permutation

Фараонууд сансрын хөлгүүдийг хурдасгахын тулд гаригуудын харьцангуй хөдөлгөөн, таталцлыг ашигладаг.
Сансрын хөлөг нь $p[0], p[1], \ldots, p[n-1]$ тойрог замын хурдтай n гаригийг дараалаар нь өнгөрнө гэж бодъё. Фараоны эрдэмтэд гариг бүрийн хувьд уг гаригийг ашиглан сансрын хөлгийг хурдасгах уу, үгүй юу гэдгээ шийдэх боломжтой. Эрчим хүч хэмнэхийн тулд тойрог замын $p[i]$ хурдтай гаригаар хурдасгасны дараа сансрын хөлөг нь тойрог замын $p[j] < p[i]$ хурдтай аль ч гаригийг ашиглан хурдасгах боломжгүй юм. Өөрөөр хэлбэл, сонгосон гаригууд нь $p[0], p[1], \ldots, p[n-1]$ дарааллаас **өсөн нэмэгдэж буй дэд дараалал** байдалтай байдаг. p-ийн дэд дараалал гэдэг нь p-ээс 0 эсвэл хэд хэдэн элемент устгаснаар p-ээс үүсэн дараалал юм. Жишээлбэл, $[0]$, $[]$, $[0, 2]$, ба $[0, 1, 2]$ нь $[0, 1, 2]$-ын дэд дараалал боловч $[2, 1]$ нь биш юм.

Эрдэмтэд сансрын хөлгийг хурдасгах олон тооны гаригийг сонгох ялгаатай k зам байдгийг тогтоосон боловч тэд тойрог замын бүх хурдны тэмдэглэлээ алдсан байна (n утгыг оролцуулан). Гэсэн хэдий ч тэд $(p[0], p[1], \ldots, p[n-1])$ нь $0, 1, \ldots, n-1$-ийн сэлгэмэл гэдгийг санаж байв. Сэлгэмэл гэдэг нь 0-ээс $n-1$ хүртэлх бүхэл тоог яг нэг удаа агуулсан дараалал юм. Таны даалгавар бол $p[0], p[1], \ldots, p[n-1]$ хангалттай бага урттай дарааллыг сэргээх явдал юм.

Та ялгаатай q сансрын хөлгийн асуудлыг шийдэх хэрэгтэй. Сансрын хөлөг i бүрийн хувьд та бүхэл тоо k_i-г авах ба энэ нь сансрын хөлгийг хурдасгах гаригуудын багцыг сонгож болох ялгаатай арга замын тоог илэрхийлнэ. Таны даалгавар бол тойрог замын хурд нь нэмэгдэж буй гаригуудын дэд дарааллыг сонгох арга зам нь яг k_i байхаар хамгийн бага n_i урттай тойрог замын хурдны дарааллыг олох явдал юм.

Хэрэгжүүлэлтийн мэдээлэл

Та дараах функцийг хэрэгжүүлэх хэрэгтэй:

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

- k: өсөн нэмэгдэж буй дэд дарааллын хүссэн тоо.
- Энэ процедур нь элемент бүр нь 0 ба $n-1$ хооронд байх n элементтэй массивыг буцаана.
- Буцаасан массив нь яг k өсөн нэмэгдэж буй дэд дараалал бүхий хүчинтэй сэлгэлт байх ёстой.
- Энэ процедурыг нийт q удаа дуудна. Дуудлага тус бүрийг тусдаа хувилбар болгон авч үзэх хэрэгтэй.

Хязгаарлалт

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

Дэд бодлого

1. (10 оноо) $2 \leq k_i \leq 90$ (for all $0 \leq i \leq q-1$). Хэрэв таны ашигласан бүх сэлгэмэл хамгийн ихдээ 90 урттай ба зөв бол та 10 оноо авна, үгүй бол 0.
2. (90 оноо) Нэмэлт хязгаарлалт байхгүй. Энэ дэд бодлогын хувьд m-ийг аль ч хувилбарт ашигласан хамгийн их сэлгэх урта гэе. Энэ үед таны оноог дараах хүснэгтийн дагуу тооцоолж өгнө:

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

Жишээ

Жишээ 1

Дараах дуудлагыг анхаарч үзье:

```
construct_permutation(3)
```

Энэ процедур нь яг 3 өсөн нэмэгдэж буй дэд дараалал бүхий сэлгэлтийг буцаах ёстой. Боломжит хариулт нь $[1, 0]$ ба $[]$ (хоосон дэд дараалал), $[0]$ ба $[1]$ нь өсөн нэмэгдэж буй дэд дарааллаар байна.

Жишээ 2

Дараах дуудлагыг анхаарч үзье:

```
construct_permutation(8)
```

Энэ процедур нь яг 8 өсөн нэмэгдэж буй дэд дараалал бүхий сэлгэлтийг буцаах ёстой. Боломжит хариулт нь $[0, 1, 2]$.

Жишээ грейдэр

Жишээ грейдэр нь дараах форматаар оролтыг уншина:

- line 1: q
- line $2+i$ $(0 \le i \le q-1)$: k_i

Түүврийн зэрэглэгч нь `construct_permutation`-ийн буцаасан утгын ганц мөр, эсвэл алдаа гарсан бол алдааны мэдэгдлийг хэвлэнэ.