Hack! (hack)

Անցել է մեկ ժամ Codeforces մրցույթից, երբ նկատում ես, որ մեկ այլ մասնակից քո սենյակում խնդիր է լուծել օգտագործելով `unordered_set`: Ժամանակն է `hack` անել:

Դու գիտես, որ unordered set-ը օգտագործում է հեշ աղյուսակ՝ n զամբյուղներով (bucket-ներով), որոնք համարակալված են 0-ից մինչև $n-1$: Ցավոք, դու չգիտես n-ի արժեքը և ցանկանում ես վերականգնել այն:

Երբ ամբողջ թիվ x-ը ներդրվում է հեշ աղյուսակի մեջ, այն ընկնում է $(x \bmod n)$ -րդ զամբյուղ: Եթե այդ զամբյուղում դրանից առաջ եղել է b տարր, ապա տեղի կունենա b հեշ բախում (collision):

Տրամադրելով k տարբեր ամբողջ թվեր՝ $x[0], x[1], \ldots, x[k-1]$ ինտերակտորին, կարող ես իմանալ ընդհանուր բախումների քանակը, որոնք տեղի են ունեցել այդ թվերով բազմություն ստեղծելիս: Սակայն k թվերով մեկ հարցում կատարելը կարժենա k միավոր:

Օրինակ, եթե $n = 5$, ապա interactor-ին փոխանցելով $x = [2, 15, 7, 27, 8, 30]$՝ կստացվի ընդհանուր 4 բախում.

Գործողություն	Նոր բախումներ	Զամբյուղների վիճակ
սկզբնական վիճակ	$-$	$[], [], [], [], []$
ներդնել $x[0] = 2$	0	$[], [], [2], [], []$
ներդնել $x[1] = 15$	0	$[15], [], [2], [], []$
ներդնել $x[2] = 7$	1	$[15], [], [2, 7], [], []$
ներդնել $x[3] = 27$	2	$[15], [], [2, 7, 27], [], []$
ներդնել $x[4] = 8$	0	$[15], [], [2, 7, 27], [8], []$
ներդնել $x[5] = 30$	1	$[15, 30], [], [2, 7, 27], [8], []$

Նշենք, որ interactor-ը ստեղծում է հեշ աղյուսակը՝ հերթականությամբ ներդնելով տարրերը դատարկ բազմության մեջ, և յուրաքանչյուր հարցման համար ստեղծվում է նոր դատարկ բազմություն: Այսինքն՝ բոլոր հարցումները անկախ են միմյանցից:

Քո խնդիրն է պարզել զամբյուղների n քանակը՝ օգտագործելով առավելագույնը $1;000;000$ ընդհանուր արժեքով հարցումներ։

Իրականացման մանրամասներ

Դու պետք է իրականացնես հետևյալ ֆունկցիան.

```
int hack()
```

- Այս ֆունկցիան պետք է վերադարձնի մեկ ամբողջ թիվ՝ գաղտնի n արժեքը։
- Յուրաքանչյուր թեստի համար գնահատիչը կարող է մի քանի անգամ կանչել այս ֆունկցիան։ Ամեն կանչ պետք է դիտվի որպես նոր, անկախ սցենար։

Այս ֆունկցիայի ներսում կարող ես կանչել հետևյալ ֆունկցիան.

```
long long collisions(std::vector<long long> x)
```

- x: տարբեր ամբողջ թվերից կազմված զանգված, որտեղ $1 \le x[i] \le 10^{18}$։
- Այս ֆունկցիան վերադարձնում է բախումների քանակը, որոնք առաջանում են x տարրերով unordered set ստեղծելիս։
- Այս ֆունկցիան կարելի է կանչել բազմիցս։ Մեկ `hack()` կանչի ընթացքում բոլոր `collisions()` կանչերի x զանգվածների երկարությունների գումարը չպետք է գերազանցի $1;000;000$։

Նշում. **քանի որ `hack()` ֆունկցիան կանչվելու է մեկից ավելի անգամ, մասնակիցները պետք է ուշադիր լինեն նախորդ կանչերից մնացած տվյալների ազդեցության նկատմամբ, հատկապես՝ գլոբալ փոփոխականներում պահված արժեքների վրա։**

$1;000;000$ արժեքի սահմանափակումը գործում է յուրաքանչյուր թեստի համար առանձին։ Ընդհանուր առմամբ, եթե կան `hack()`-ի t կանչեր, ապա կարելի է օգտագործել ընդհանուր $t \times 1;000;000$ արժեք, այնպես որ յուրաքանչյուր `hack()` կանչ չգերազանցի $1;000;000$։

Interactor-ը ադապտիվ չէ (non-adaptive), այսինքն՝ բոլոր թեստերի n արժեքները նախապես որոշված են։

Օրինակ

Ենթադրենք՝ կա 2 մուլտիթեստ։ Գնահատիչը կանչում է՝

```
hack()
```

Ենթադրենք՝ ֆունկցիայի ներսում կատարում ես հետևյալ հարցումները.

Կանչ	Վերադարձված արժեք
`collisions([2, 15, 7, 27, 8, 30])`	4
`collisions([1, 2, 3])`	0
`collisions([10, 20, 30, 40, 50])`	10

Եթե հայտնաբերում ես, որ $n = 5$, ապա `hack()` պետք է վերադարձնի 5։

Այնուհետև գնահատիչը նորից կանչում է՝

```
hack()
```

Ենթադրենք դու կատարում ես՝

Կանչ	Վերադարձված արժեք
`collisions([1, 3])`	1
`collisions([2, 4])`	1

Միայն $n = 2$ է բավարարում այս արդյունքներին։ Ուրեմն `hack()` պետք է վերադարձնի 2։

Սահմանափակումներ

- $1 \leq t \leq 10$, որտեղ t-ն մուլտիթեստերի քանակն է։
- $2 \leq n \leq 10^9$
- $1 \leq x[i] \leq 10^{18}$ յուրաքանչյուր `collisions()` կանչի համար։

Ենթախնդիրներ

1. (8 միավոր) $n \leq 500;000$

2. (17 միավոր) $n \leq 1;000;000$

3. (75 միավոր) Լրացուցիչ սահմանափակումներ չկան։

Վերջին ենթախնդիրում կարող եք ստանալ մասնակի միավոր։ Թող q լինի բոլոր `hack()` կանչերի ընթացքում օգտագործված առավելագույն արժեքը։ Քո միավորը հաշվարկվում է հետևյալ աղյուսակի համաձայն.

Պայման	Միավորներ
$1;000;000 < q$	0
$110;000 < q \leq 1;000;000$	$75 \cdot \log_{50}\left(\frac{10^6}{x-90000}\right)$
$q \leq 110;000$	75

Եթե որևէ թեստում `collisions()` կանչերը չեն համապատասխանում իրագործման պայմաններին, կամ `hack()`-ը սխալ արժեք է վերադարձնում, այդ ենթախնդրի միավորը կլինի 0։

Նմուշային գնահատիչ

Նմուշային գնահատիչը մուտքը կարդում է հետևյալ ձևաչափով՝

* տող 1: t

Այնուհետև հաջորդում են t տողեր՝ յուրաքանչյուրում n արժեքը.

* տող 1: n

Յուրաքանչյուր թեստի համար՝ եթե m-ը `hack()` վերադարձված արժեքն է, իսկ c-ն՝ հարցումների արժեքի գումարը, գնահատիչը տպում է.

* տող 1: $m; c$