



НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 12 – 14 юни 2026 г.

Група В, 9 – 10 клас

Задача В1. ЛОВ

 1.5 сек.  1024 MB

Елмър Фъд е на лов за зайци. По-конкретно, той, както винаги, иска да залови Бъгс Бъни. Бъгс се крие из една кръгла гора, която разглеждаме като циклична лента, разделена на L сектора (номерирани от 0 до $L - 1$), където за съседни считаме сектори i и $(i + 1) \bmod L$ за $0 \leq i < L$.

Елмър знае, че в ден 0 Бъгс се е намирал в един от N различни сектора: P_0, \dots, P_{N-1} . Също така, всяка вечер, включително тази на ден 0, Бъгс се мести от текущия си сектор към един от съседните: от P към $(P - 1) \bmod L$ или $(P + 1) \bmod L$. Забележете че, той никога не остава на място.

Елмър пристига в гората сутринта на ден H (където $H > 0$) и от там насетне, всяка сутрин, включително тази на ден H , той поставя капан в един от секторите, който той си избира. Ако Бъгс някога се окаже в сектор с капан (било то през деня веднага след поставянето или влизайки в сектора някоя вечер), той бива заловен.

Целта на Елмър е да гарантира, че ще залови Бъгс рано или късно, като освен това иска да използва възможно най-малко капани. Обърнете внимание, че е възможно Бъгс да бъде заловен в по-късен ден от поставянето на последния капан; важно е само това да се случи някога.

Помогнете на Елмър, като напишете програма **hunt**, която по дадени L , H и списък от възможни начални сектори P_0, \dots, P_{N-1} , открива минималния възможен брой капани нужни на Елмър, за да залови Бъгс със сигурност (като поставя по един капан на ден).

Детайли по имплементацията

Трябва да имплементирате функцията `solve`:

```
long long solve(long long L, long long H, std::vector<long long> P)
```

- L : Броят сектори в гората.
- H : Денят, в който Елмър пристига; т.е. първият ден, в който той слага капан.
- P : Списъкът от възможни начални сектори.

Тази функция ще бъде извикана точно веднъж в рамките на едно изпълнение на програмата и трябва да върне едно число: минималният нужен брой капани.

Ограничения

- $1 \leq N \leq 2 \times 10^6$
- $1 \leq H \leq 10^{12}$
- $1 \leq L \leq 10^{15}$
- $0 \leq P_i < L$
- $P_i \neq P_j$ за $i \neq j$
- $N \leq L$



НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 12 – 14 юни 2026 г.

Група В, 9 – 10 клас

Подзадачи

№	Точки	Необходими подзадачи	N	Други ограничения
0	0	–	–	Примерни тестове
1	3	–	$= 1$	$P_i + 2H + 4N < L$
2	3	1	$= 1$	–
3	3	1	≤ 2	$P_i + 2H + 4N < L$
4	3	1 – 3	≤ 2	–
5	4	1, 3	≤ 4	$P_i + 2H + 4N < L$
6	4	1 – 5	≤ 4	–
7	4	1, 3, 5	≤ 15	$P_i + 2H + 4N < L$
8	4	0 – 7	≤ 15	–
9	7	1, 3, 5, 7	≤ 100	$P_i + 2H + 4N < L$
10	7	0 – 9	≤ 100	–
11	14	0 – 10	≤ 420	–
12	14	0 – 11	≤ 9000	–
13	15	0 – 12	$\leq 10^5$	–
14	15	0 – 13	$\leq 2 \times 10^6$	–

Примери

Вход	Изход
1 7 1	3
1	

Вход	Изход
5 969 44	237
4 108 619 887 408	

В първия тест $N = 1$, $L = 7$, $H = 1$ и единствената възможна начална позиция е $P_0 = 1$. Възможно е решение с 3 капана, което е и оптимално. Едно примерно такова е да се сложат следните капани: в сектор 2 в ден 1, в сектор 6 в ден 2 и в сектор 0 в ден 3. Нека го проследим:

- Вечерта на ден 0, Бъгс се мести от 1 към 0 или 2.
- Сутринта на ден 1, Елмър поставя капан в 2. Така, ако заека е там ще го хване и остава да се погрижи за ситуацията, в която е в 0.
- Вечерта на ден 1, Бъгс се мести от 0 към 6 или 1.
- Сутринта на ден 2, Елмър поставя капан в 6. Аналогично на ден 1 остава случаят заекът да е в 1.
- Вечерта на ден 2, Бъгс се мести от 1 към 0 или към 2, но ако се премести към 2 ще бъде заловен веднага и остава да се справим с това ако се е преместил в 0.
- Сутринта на ден 3, Елмър поставя капан в 0, което е единствената оставаща възможност. Така гарантира, че залавя Бъгс.

Обърнете внимание, че в този пример Бъгс бива заловен най-късно в ден 3, но това не е от значение. Важно е само броят дни, в които Елмър поставя капани, като е възможно Бъгс да бъде заловен и по-късно.



НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Пловдив, 12 – 14 юни 2026 г.

Група В, 9 – 10 клас

Локален грейдър

Формат на входа:

- ред 1: три числа: N, L, H ;
- ред 2: N числа: P_0, \dots, P_{N-1} .

Формат на изхода:

- ред 1: върнатата стойност при извикването на `solve`.