

**ПЪРВО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 21 АПРИЛ, 2019 Г.
ГРУПА А**

Задача АК1. Lowest Hash

Наскоро Ели, Крис и Станчо научиха следния алгоритъм за хеширане на стринг S с дължина N , съставен от главни букви на английската азбука:

```
long long hash = 0;
for (int i = 0; i < N; i++) {
    hash = (hash * 127 + S[i]) % 10000000000000037;
}
```

$S[i]$ дава ASCII кода на i -тата буква от стринга. Буквата 'A' е с ASCII код 65, докато 'Z' - с код 90. Стойността на променливата $hash$ след края на цикъла се нарича "хеш" на стринга.

Всеки от тримата има по един любим стринг с дължина N : този на Ели е **A**, този на Крис е **B**, а този на Станчо е **C**. Сега те са решили да съставят стринг S със същата дължина, който на всяка от позициите си съдържа една от буквите на съответната позиция в някой от техните стрингове. С други думи, за позиция i , $S[i] = A[i]$ или $S[i] = B[i]$ или $S[i] = C[i]$. Ели, Крис и Станчо искат да намерят стринга S с най-малък хеш. Можете ли да им помогнете, като напишете програма **LowestHash**, която го намира вместо тях?

Вход

На първият ред на стандартния вход ще бъде зададено цялото число N - дължината на всеки от стринговете. На всеки от следващите три реда ще бъде зададен по един стринг с дължина N , съставен от главни букви от английската азбука - съответно стринговете на Ели (**A**), Крис (**B**), и Станчо (**C**).

Изход

На единствен ред на стандартния изход изведете едно цяло число - минималния възможен хеш, който може да има стринг S , образуван по описания по-горе начин.

Ограничения

❖ $1 \leq N \leq 28$

<i>Примерен Вход</i>	<i>Примерен Изход</i>
14 ELEONORAEKIFLA KRISIMASTRING STANCHORIEVODA	545601337476

Примерни стрингове, които могат да съставят, са STISNIMASTRONG (с хеш 816860215893321) или пък ETENHRPEEIOA (с хеш 881156714759922) или дори KRISIMASTRING (с хеш 515697493951910), но този с най-малък такъв е KTESNOORPEEIFDA (с хеш 545601337476).

**ПЪРВО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 21 АПРИЛ, 2019 Г.
ГРУПА А**

Задача АК2. ТРИЪГЪЛНИЦИ

YouTube-ката Vi Hart е известна с любовта си към триъгълниците. Тя толкова ги обича, че иска да си нарисова възможно най-много различни триъгълници. Сега тя има един набор от N отсечки с различни цели дължини. От тях тя иска да състави всички възможни валидни триъгълници (ъглите им да са по-големи от 0° и по-малки от 180°), като ротации и симетрии не се броят за различни триъгълници, т.е. значение има само какви са дължините на страните (една отсечка може да се използва повече от веднъж).

За целта Vi решила да си напише програма, която по зададен набор от отсечки да отговаря колко различни валидни триъгълника могат да се получат. За жалост обаче не се справила и написала само входната и изходната функция. Помогнете ѝ като напишете основаната част от програмата – функция *triangles*, която да се компилира с нейната програма и да отговаря на задачата.

Детайли по имплементацията

Функция *triangles* трябва да има следния прототип:

```
long long triangles(const int lens[], int n);
```

Тя ще бъде извикана точно веднъж и ще получи като аргументи масив с дължините на отсечките и бройката им.

Към системата трябва да изпратите файл **triangles.cpp**, в който трябва да е имплементирана функцията Ви. В него може да имате каквито искате помощни функции, структури, променливи и т.н. Той само не трябва да съдържа функция *main* и трябва да включва хедър файла **triangles.h** чрез указание към препроцесора `#include "triangles.h"` в началото.

Ограничения

$$1 \leq N \leq 340\,000$$

$$1 \leq \text{дължини на отсечките} \leq \lfloor 3N/2 \rfloor$$

Подзадачи и оценяване

За да получите точките за дадена подзадача, решението Ви трябва успешно да премине всички тестове в нея.

Подзадача 1 (10 точки): $N \leq 1\,750$

Подзадача 2 (15 точки): $N \leq 8\,000$

Подзадача 3 (15 точки): $N \leq 22\,500$

Подзадача 4 (10 точки): $N \leq 53\,000$

Подзадача 5 (25 точки): $N \leq 130\,000$

Подзадача 6 (25 точки): $N \leq 340\,000$

**ПЪРВО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 21 АПРИЛ, 2019 Г.
ГРУПА А**

Локално тестване

Предоставени Ви са файловете **triangles.h** и **Lgrader.cpp**, които можете да компилирате заедно с Вашата програма, за да я тествате.

При стартиране програмата ще Ви пита за N , последвано от дължините на отсечките. Ако искате да я конфигурирате по друг начин, може да правите каквито си промени искате по предоставените Ви файлове.

Примерен тест

Вход	Изход
4 3 6 1 4	13

Обяснение на примерния тест

Валидните триъгълници са следните:

1 1 1
1 3 3
1 4 4
1 6 6
3 3 3
3 3 4
3 4 4
3 4 6
3 6 6
4 4 4
4 4 6
4 6 6
6 6 6

**ПЪРВО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 21 АПРИЛ, 2019 Г.
ГРУПА А**

Задача АКЗ. ГОБЛИНИ

В света, в който живее Goblin Slayer (Убиеца на гоблини), въпросните същества живеят в пещери. Пещерите са N на брой и са разположени по билото на една дълга планина. Когато Goblin Slayer посети някоя пещера, той убива всички гоблини в нея, но тъй като те бързо се размножават, същата бройка гоблини ще се появят отново в пещерата на по-следващия ден (докато на следващия, тя все още ще е празна). Целта на нашия протагонист е да убие възможно най-много гоблини за даден брой дни. Всеки ден той посещава по една пещера и след като приключи лагерува пред нея за нощта. На следващия ден той може да посети или предната или следващата пещера, т.е. ако на ден t той е посетил пещера i , на ден $t + 1$ той може да посети или $i - 1$ или $i + 1$.

В началото на ден 1, Goblin Slayer се намира на позиция 0 (т.е. позицията от ляво на първата пещера, която се намира на позиция 1) и всички пещери са пълни, като в пещера i има G_i гоблина.

Преди да започне своята кампания той обаче още не е сигурен колко дни ще продължи тя. Помогнете му да реши, като напишете програма **goblins.cpp**, която отговаря на Q на брой заявки. Всяка заявка пита колко е максималният брой гоблини, които той може да убие, ако остане T_j дни.

Вход

От първия ред на стандартния вход се въвеждат две цели положителни числа N и Q – броя пещери по планината и броя заявки. На следващия ред се въвеждат N цели положителни G_1, G_2, \dots, G_N – бройките гоблини в пещерите. На следващите Q реда се въвеждат заявките подред, всяка описана от по едно число T_j – бройката дни.

Изход

На Q реда на стандартния изход програмата трябва да изведе едно цяло положително число – отговорите на заявките в реда, в който те са зададени.

Ограничения

$$2 \leq N, Q \leq 10^6$$

$$1 \leq G_i, T_j \leq 10^7$$

$$T_j \neq T_k \text{ ако } j \neq k$$

Подзадачи и оценяване

За да получите точките за дадена подзадача, решението Ви трябва успешно да премине всички тестове в нея.

Подзадача 1 (5 точки): $N, Q, T_j \leq 10$

Подзадача 2 (10 точки): $N, Q, T_j \leq 15\,000$

Подзадача 3 (10 точки): $N, Q \leq 18\,000$

Подзадача 4 (20 точки): Стойностите на G_i и T_j са произволно генерирани.

Подзадача 5 (55 точки): Без допълнителни ограничения.

**ПЪРВО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 21 АПРИЛ, 2019 Г.
ГРУПА А**

Примерен тест

Вход	Изход
4 3	7
3 1 2 3	3
3	11
1	
5	

Обяснение на примерния тест

Оптималната кампания за три дни е през следните пещери: $1 \rightarrow 2 \rightarrow 1$.

При дължина само един ден единствената възможна кампания е: 1

Оптималната кампания за четири дни е през следните пещери: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 3$.