

А есебі. Жартылай бос па, жартылай толы ма?

Енгізу файлының аты: A.in
Шығару файлының аты: A.out
Уақыт шектеу: 1 секунд
Жадыға шектеу: 256 мегабайт

Жандос, Алан, Нұрдаулет, Елібай және Жарасхан тәжірибелі спорт бағдарламашылары. Әр дайындықтан кейін олардың бір-бір стақан Коладан ішетін дәстүрі бар. Олардың ішінде Жандос ең үлкені болғандықтан, бір бөтелкені толығымен бәріне бөліп, құйып шықты. Алайда, Нұрдаулет мұқият математик болғандықтан, кейбір стақандардағы Коланың көлемі тең емес екенін байқап қойды. Олар ақылдасып, әр стақандағы Коланың көлемін стақаннан-стақанға құю арқылы теңестіруге шешім қабылдады. Бір рет құю операциясы — кез келген адам басқа кез келген адамды таңдап, оның стақанына Коланы қалаған көлемінде өзінің стақанынан құйып бере алады. Әрине, өзінде бар көлемінен көп құйып бере алмайды және құйған Коланың көлемі миллилитрмен **бүтін** сан болу керек. Стақандардың сыйымдылығы жеткілікті.

Бұл бағдарламашылар еріншек болғандықтан, мүмкіндігінше аз құю операцияларын жұмсағысы келеді. Сонымен қатар, **2-ден көп операция жұмсамай** теңестіргісі келеді. Әр стақанның ішінде бар Коланың көлемін біле тұра, 1 немесе 2 операциямен теңестіру мүмкіндігін тексеріңіз. Мысалдардың түсіндіріліміне назар аударыңыз.

Енгізу файлының форматы

Бірінші жолда оң бүтін сан T ($1 \leq T \leq 10$) — сынақ жағдайлар саны берілген. Келесі T жолда 5 оң бүтін сан $1 \leq A, B, C, D, E \leq 10^6$ — Жандос, Алан, Нұрдаулет, Елібай және Жарасханның стақанындағы Коланың көлемі миллилитрмен берілген. Әр сынақ жағдайы үшін Коланың көлемі тең емес 2 стақанның бар екендігі белгілі.

Шығару файлының форматы

Әр сынақ жайғайының жауабын жеке бір жолға шығарыңыз. Бір операциямен теңестіруге мүмкін болса, «1» (тырнақшасыз) жолын шығарыңыз. Екі операция арқылы мүмкін болса, «2» (тырнақшасыз) жолын шығарыңыз. 2-ден көп операция керек болса немесе теңестіру мүмкін болмаса, «NO» (тырнақшасыз) деп шығарыңыз.

Мысал

A.in	A.out
4	1
1 2 2 2 3	NO
1 2 2 2 4	2
8 1 3 4 4	NO
9 1 3 3 4	

Түсініктеме

Тесттердің 40%-ы үшін: $1 \leq A, B, C, D, E \leq 10$.

Бірінші жағдайда Жарасхан 1 мл Коланы Жандостың стақанына құйып берсе, бәрінде 2 мл-ден болады. Екінші жағдайда теңестіру тіпті мүмкін емес. Үшінші жағдайда Жандос 3 мл Аланға және 1 мл Нұрдаулетке құйса, бәрінде 4 мл-ден болады. Төртінші жағдайда теңестіру үшін 3 операция керек, сол үшін жауабы «NO».

Задача А. Наполовину пуст или наполовину полон?

Имя входного файла: A.in
Имя выходного файла: A.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Жандос, Алан, Нурдаулет, Елибай и Жарасхан опытные спортивные программисты. И по традиции после каждой тренировки они собираются и пьют по одному стакану *Колы*. *Жандос* как самый старший разлил по стаканам всю бутылку, но *Нурдаулет* как дотошный математик заметил, что у некоторых объем *Колы* в стакане не одинаковый. Посоветовавшись они решили уравнивать объем *Колы* во всех стаканах посредством переливаний. За одно переливание любой участник может выбрать любого другого участника и перелить ему сколько угодно *Колы* со своего стакана, конечно же нельзя переливать больший объем чем имеется у участника и можно переливать только **целое количество миллилитров**. Объем всех стаканов можете считать достаточно большим.

Они ленивые и хотят уравнивать за минимальное количество переливаний, но в тоже время их не волнует если минимальное количество больше **двух** или вовсе невозможно уравнивать. Зная объем *Колы* в каждом стакане, помогите им узнать можно ли уравнивать за 1 переливание или за 2. **Внимательно** прочтите пояснения к примерам.

Формат входного файла

Первая строка входных данных содержит целое положительное число T ($1 \leq T \leq 10$) — количество тестовых данных. В последующих T строках входных данных содержится по 5 положительных целых чисел $1 \leq A, B, C, D, E \leq 10^6$ — объем *Колы* в миллилитрах в стаканах *Жандоса, Алана, Нурдаулета, Елибая и Жарасхана*, соответственно. Гарантируется, что существует два стакана в которых объем *Колы* неравно.

Формат выходного файла

На каждый тест выведите ответ в отдельной строке. Если возможно уравнивать за одно переливание, то выведите «1» (без кавычек). Если возможно уравнивать за два переливания, то выведите «2» (без кавычек). Иначе выведите «NO» (без кавычек).

Пример

A.in	A.out
4	1
1 2 2 2 3	NO
1 2 2 2 4	2
8 1 3 4 4	NO
9 1 3 3 4	

Примечание

В 40% тестов: $1 \leq A, B, C, D, E \leq 10$.

В первом примере *Жарасхан* наливает 1 мл *Жандосу* и у всех становится по 2 мл. Во втором тесте невозможно уравнивать вышеуказанным способом. В третьем примере *Жандос* наливает 3 мл *Алану* и 1 мл *Нурдаулету*, и у всех становится по 4 мл. В четвертом примере мы видим что можно уравнивать за 3 переливания, но нас это не интересует поэтому ответ «NO».

В есебі. Кэш-ауытқулары

Енгізу файлының аты: B.in
Шығару файлының аты: B.out
Уақыт шектеу: 1 секунд
Жадыға шектеу: 64 мегабайт

Сіздің досыңыз массив (деп атайық) элементтерін m рет шақыратын программа құрды. Берілген массивтің жұмыс істеу жылдамдығы өте баяу болғандықтан, досыңыз сізді тезірек істейтін b массиві арқылы жасалған кэш құруды өтінді.

Кэштің жұмыс істеу тәртібі мынадай: егер сұралған элемент кэшта сақталған болса, онда оның мәні мән кэштің өзінен алыну керек (демек, кэшқа түсу). Болмаса, элемент мәні алдымен кэшқа жүктеліп барып қана қайтарылады (яғни, кэш-ауытқу).

Кэштің өлшемдеріне шектеу қойылғандықтан, белгілі бір уақытта жаңа элементке орын қалмайды. Бұл жағдайда жаңа элементті сақталғандардың біреуінің орнына жүктеуге тура келеді.

Сіз досыңыздың программасы массивінен сұрайтын элементтердің барлығын білетіндіктен, сіздің тапсырмаңыз осы сұраныстар үшін минималды кэш-ауытқулар санын анықтау болып табылады.

Енгізу файлының форматы

Енгізілетін файлдың алғашқы жолында үш бүтін сан: n — массивінің өлшемі, k — b массивінің өлшемі (кэш өлшемі) және m — массивіне жасалған сұраныстар саны берілген ($1 \leq n, k, m \leq 10^5$).

Екінші жолында m idx_i ($1 \leq idx_i \leq n$) бүтін сандары берілген — массивіндегі сұралған элемент индекстері.

Шығару файлының форматы

Тек бір ғана санды — минималды кэш-ауытқулар санын шығарыңыз.

Мысалдар

B.in	B.out
2 1 3 1 1 2	2
2 1 3 1 2 1	3
2 2 3 1 2 1	2
6 3 8 1 2 1 1 4 5 1 6	5
3 2 6 1 2 3 1 2 3	4

Задача В. Кэш-промахи

Имя входного файла: B.in
Имя выходного файла: B.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Ваш друг написал программу, которая m раз запрашивает какие-то элементы из массива (назовем его a). Так как доступ к этому массиву очень медленный, он попросил вас реализовать кэш при помощи массива b , с гораздо более быстрым доступом.

Кэш работает следующим образом: если запрашиваемый элемент уже хранится в кэше, то возвращается значение из кэша (т.е. случилось *попадание в кэш*). В противном случае значение сначала загружается в кэш, а затем загруженное значение возвращается запросившему (т.е. случился *кэш-промах*).

Так как кэш ограничен в размерах, то в какой-то момент в нем будет недостаточно места для нового элемента. В таком случае придется загрузить новый элемент вместо уже существующего.

Так как вам известны все запросы к массиву a , которые сделает программа вашего друга, то вашей основной задачей является посчитать минимальное количество кэш-промахов, которое может случиться для этих запросов.

Формат входного файла

В первой строке входного файла заданы числа n , k и m ($1 \leq n, k, m \leq 10^5$) — размер массива a , размер массива b (то есть размер кэша) и количество обращений к массиву a соответственно.

Вторая строка входного файла содержит m целых чисел idx_i ($1 \leq idx_i \leq n$) — индексы запрашиваемых элементов в массиве a .

Формат выходного файла

Выведите единственное число - минимальное количество кэш-промахов, которое может случиться для данных обращений к массиву.

Примеры

B.in	B.out
2 1 3 1 1 2	2
2 1 3 1 2 1	3
2 2 3 1 2 1	2
6 3 8 1 2 1 1 4 5 1 6	5
3 2 6 1 2 3 1 2 3	4

Примечание

В первом тесте сначала единица загрузится в кэш (кэш-промах), затем вторая единица считается из кэша (попадание в кэш) и двойка загрузится в кэш (кэш-промах).

Во втором тесте в каждое число будет загружено в кэш, поэтому при всех трех обращениях к массиву будет кэш-промах.

В третьем тесте оба запрашиваемых значения помещаются в кэш, поэтому будет только два кэш-промаха — когда первый и второй элементы впервые попадут в кэш.

С есебі. Карталар

Енгізу файлының аты: C.in
Шығару файлының аты: C.out
Уақыт шектеу: 2 секунд
Жадыға шектеу: 64 мегабайт

Үстелдің үстінде n карта жатыр. Әр қайсысының үстінгі және астыңғы беттерінде қосындысы 1-ге тең сан екі сан жазылған. Сіздің тапсырмаңыз әр картада бір-бір саннан солай таңдау, үстінгі бетіндегі таңдалған сандардың қосындысы X -тен аспау керек тура солай астындағы таңдалғандардың саны берілген X -тен аспау керек.

Енгізу файлының форматы

Бірінші қатарда карталардың саны n берілген ($1 \leq n \leq 10^5$) және X берілген ($0 \leq X \leq n$).

Екінші қатарда n сан жазылған әр қайсысы 0 ден 1 ге дейін.

Шығару файлының форматы

Егер тапсырма орындау мүмкін болмаса -1 шығарыңыз.

Тапсырма орындай алсаңыз бірінші қатарда үстінгі саны таңдалған карталардың санын шығарыңыз. Ал екінші қатарда олардың нөмірін шығарыңыз.

Мысалдар

C.in	C.out
5 1.5 0.3 0.2 0.5 0.5 0.1	4 1 2 3 5
6 0.5 0.5 0.5 0.5 0 0 0	-1

Түсініктеме

Тесттердің 20 пайызында $n \leq 20$ Тағы 30 пайызында $n \leq 1000$

Бірінші тестте, үстінде таңдалған сандардың қосындысы $0.1 + 0.2 + 0.3 + 0.5 < (5 + 1)/4$. Ал астындағы таңдалған сандардың қосындысы $0.5 \leq (5 + 1)/4$.

Екінші тестте тапсырма орындау мүмкін емес.

Задача С. Карты

Имя входного файла: C.in
Имя выходного файла: C.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На столе лежат n карт, на каждой стороне написано по вещественному числу сумма которых 1. Ваша задача выбрать на каждой карте по числу так чтобы сумма выбранных чисел на верхней части и сумма выбранных чисел на обратной части была не больше X

Формат входного файла

В первой строке дано количество карт n ($1 \leq n \leq 10^5$.) и число X ($0 \leq X \leq n$)
Во второй n чисел через пробел - числа на верхней стороне карт. Все числа от 0 до 1.

Формат выходного файла

Если нет ответа выведите 1"без кавычек.

В первой строке одно число - количество чисел которые вы выберете на верхней части.

Во второй строке индексы выбранных чисел у которых выбрана верхняя сторона.

Если ответов несколько разрешается вывести любой из них. Ответ будет считаться правильным, если его относительная или абсолютная погрешность не превышает 10^{-4} .

Примеры

C.in	C.out
5 1.5 0.3 0.2 0.5 0.5 0.1	4 1 2 3 5
6 0.5 0.5 0.5 0.5 0 0 0	-1

Примечание

20% тестов имеет $n \leq 20$.

Еще 30% тестов имеет $n \leq 1000$.

В первом тесте если выбрать верхние части карт номером 1,2,3,5 сумма выбранных чисел на верхней части будет $0.1 + 0.2 + 0.3 + 0.5 \leq (5 + 1)/4$ также $0.5 \leq (5 + 1)/4$ для нижней части.

Во втором тесте невозможно выбрать карты так чтобы сумма на обеих сторонах не превысила 0.5.

D есебі. Жаңа жыл шыршасы

Енгізу файлының аты: D.in
Шығару файлының аты: D.out
Уақыт шектеу: 1 секунд
Жадыға шектеу: 64 мегабайт

НурлашКО жаңа жыл қарсаңында бір төбеден (номері 1) тұратын екілік ағаш сатып алды. Ол осы ағаштан жаңа жыл шыршасын өсіруге шешім қабылдады. Шырша деп — барлық жапырақтары бірдей биіктікте орналасқан, ал басқа төбелерінің барлығынан 2 бұтақ немесе 2 жапырақ өсіп тұрған, екілік ағашты атайды.

Екілік ағаштарды өсіру өте оңай. Нөмері V төбені суғарса, осы төбеден басталатын бұтақтың әрбір жапырағынан жаңа екі $2*leaf$ және $2*leaf+1$ жапырақтары өсіп шығады, $leaf$ — жапырақтың нөмері.

НурлашКО ағашты суғару үшін N бірлік сұйық дайындап алды. Екілік ағаштар тез өсетіндіктен және өсуі тұрақсыз болғандықтан, *НурлашКО* төбелерді бірінен соң бірін суғара бастады. Ол бұл істі болғаннан кейін, кездейсоқ төбелерді суғару дұрыс ой болмағанын түсінді. Өйткені, өскен ағашы шырша болмай қалуы мүмкін. Қуанышқа орай, екілік ағашта, кез келген төбеден бастап бұтағын кесіп алуға болады.

Ендігі қалыптасқан сұрақ — қандай ең көп төбеден тұратын шырша кесіп алуға болады?

Енгізу файлының форматы

Бірінші жолда бүтін сан N ($1 \leq N \leq 10^5$) — су мөлшері берілген. Екінші жолда N бүтін сан V_1, V_2, \dots, V_n ($1 \leq V_i \leq 10^{18}$) берілген. Қатардағы i -інші сан i -інші болып суғарылған төбенің нөмері. Кез келген V төбесін суғару кезінде, бұл төбенің ағашта әлдеқашан бар екені белгілі.

Шығару файлының форматы

Жауапқа бір сан шығарыңыз — шырша бола алатын ең үлкен бұтақтың бойындағы төбелердің саны. Жауабы өте үлкен бола алатындықтан, оны $10^9 + 7$ санына бөлгендегі қалдықты шығарыңыз.

Бағалау жүйесі

20% тесттерде: $1 \leq N \leq 10$.

20% тесттерде: $1 \leq V_i \leq 3$.

Мысалдар

D.in	D.out
2 1 1	7
3 1 1 4	3

Түсініктеме

Төменде екінші мысалдағы ағаш көрсетілген.

Жасылмен ең үлкен шырша бұтақтар белгіленген. Сонымен қатар, 5, 6, 7, 8, 9 төбелерінен басталатын бұталар шырша болып саналады, бірақ, олардағы төбелер саны біреу ғана. 1 және 2 төбелерінен басталатын бұталар шырша бола алмайды, сондықтан, олар жауап бола алмайды.

Задача D. Новогодняя ёлка

Имя входного файла: D.in
Имя выходного файла: D.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В преддверии нового года НурлашКО приобрел в магазине бинарное дерево из одной вершины с номером 1. Он решил вырастить из этого дерева новогоднюю елку. *Ёлкой* будем называть идеальное бинарное дерево, то есть такое дерево в котором все листовые вершины расположены на одинаковой высоте, а не листовые имеют ровно по два сына.

Растить бинарные деревья очень просто. Если полить какую либо вершину V , то из каждого листа в поддереве вершины V вырастает по две новые вершины с номерами $2 * leaf$ и $2 * leaf + 1$, где $leaf$ — номер листа.

Для полива НурлашКО приготовил N единиц жидкости. Бинарные деревья растут быстро и во время роста очень не стабильны, поэтому было решено поливать вершины одна за другой. После того как процесс полива закончился стало ясно что поливание случайных вершин было не лучшей идеей, теперь полученное дерево не обязательно является *ёлкой*. К счастью у бинарных деревьев без особого вреда для самого дерева можно один раз вырезать вершину вместе со всем ее поддеревом.

Теперь осталось только узнать какого максимального по количеству вершин размера можно вырезать из данного дерева.

Формат входного файла

Первая строка входного файла будет содержать одно число N ($1 \leq N \leq 10^5$) — количество воды.

Во второй строке записаны N целых чисел V_1, V_2, \dots, V_n ($1 \leq V_i \leq 10^{18}$). i -ое число в списке является номером вершины которая была полита i -ой по порядку. Гарантируется, что на момент когда ваша программа считает вершину V она уже будет находится в дереве.

Формат выходного файла

Выведите одно число — количество вершин в наибольшем поддереве являющейся ёлочкой. Так как это число может быть очень большим выведите его по модулю $10^9 + 7$.

Система оценки

В 20% тестовых данных: $1 \leq N \leq 10$.

В еще 20% тестовых данных: $1 \leq V_i \leq 3$.

Примеры

D.in	D.out
2 1 1	7
3 1 1 4	3

Примечание

Ниже показано дерево из второго примера.

Зеленым выделены ёлки с максимальным количеством вершин.

Так же под ёлки подходили подеревья в вершинах с номерами 5, 6, 7, 8, 9, но они содержат всего одну вершину.

Поддеревья в вершинах с номерами 1 и 2 не подходят так как не являются ёлками.



