

Problem Picasso. Зебри

Input file: input.txt or standard input
Output file: output.txt or standard output
Time limit: 1 second
Memory limit: 512 megabytes

Олег си води дневник, където отбелязва всеки ден, дали е хубав или лош. Той нарича *зебра* непразна редица от дни, започваща и завършваща на лош ден, в която хубавите и лошите дни се редуват. Нека означим лошите дни с 0, а хубавите с 1. Тогава например редиците 0, 010, 01010 са зебри, а редиците 1, 0110 и 0101 не са.

Олег ви е разказал историята на живота си в хронологичен ред, като редица от 0 и 1, и сега вие се интересувате, може ли да разбиете историята на Олег на няколко **подредици**, всяка от които е зебра, и, ако да, то как това може да се направи. Всеки ден трябва да попадне в точно една подредица. Дните във всяка подредица трябва да са подредени в хронологичен ред. Обърнете внимание, че подредицата не е задължително да съдържа група от последователни дни.

Input

На единствения ред на стандартния вход е записан непразен низ s , състоящ се от 0 и 1, който описва дневника на Олег. Дължината на низа (обозначаваме я като $|s|$) не надвишава 200 000 символа.

Output

Ако съществува начин да се разбие дневника на Олег на подредици, които са зебри, то на първия ред изведете число k ($1 \leq k \leq |s|$) — броя на подредиците. На всеки i -ти от следващите k реда, изведете първо цяло число l_i ($1 \leq l_i \leq |s|$) — дължината на i -тата подредица, а след това l_i числа — номерата на дните, от съответната подредица. Номерата трябва да са подредени във възходящ ред, като първия ден е с номер едно. Всеки номер от 1 до n трябва да попада точно в една подредица. Ако не съществува разбиване на подредици-зебри изведете -1.

Подредиците може да изведете в произволен ред. Ако са възможни няколко разбивания на подредици-зебри, то изведете коя да е от тях. Не се изисква да намерите минимална или максимална стойност на k .

Examples

input	output
0010100	3 3 1 3 4 3 2 5 6 1 7
111	-1

Scoring

Тестовите са разделени на три групи. Точките за всяка група получавате само ако работят всички тестове от групата и всички тестове от предишните групи.

Група	Точки	Дополнителни ограничения	Коментари
		$ s $	
0	0	—	Тестовите от условието
1	31	$ s \leq 10$	—
2	29	$ s \leq 2000$	—
3	40	—	—

Problem Kandinsky. Обновяване на data-центрове

Input file: input.txt or standard input
Output file: output.txt or standard output
Time limit: 1 second
Memory limit: 512 megabytes

В компанията BigData Inc. има n data-центъра, номерирани от 1 до n , разположени по целия свят. В тези data-центрове се съхраняват данни на клиенти на компанията (както може да се досетим от името — голям обем от данни!)

Основен приоритет на предлаганите от компанията BigData Inc. услуги е гарантираната възможност за работа с потребителските данни, даже при условие, че до някой от data-центровете на компанията няма достъп. Подобна гаранция се достига чрез използване на *двойно дублиране* на данните. Двойно дублиране е подход, при който всички данни се съхраняват в две идентични копия в два различни data-центъра.

За всеки от m клиента на компанията са известни номерата на двата различни data-центъра $c_{i,1}$ и $c_{i,2}$, в които се съхраняват неговите данни.

За поддържане на работоспособността на data-центровете и безопасността на данните, програмното осигуряване на всеки data-център изисква редовно обновяване. Релийзния цикъл в компанията BigData Inc. е един ден, т.е. новата версия на програмното осигуряване се качва на всеки компютър на data-центъра всеки ден.

Обновяването на data-център, съставен от много компютри е сложна и продължителна задача, затова за всеки data-център е отделен времеви интервал с продължителност един час, в течение на който компютрите от data-центъра се обновяват и като следствие от това могат да бъдат недостъпни. Приемаме, че в денонощието има h часа. Тогава за всеки data-център е фиксирано цяло число u_j ($0 \leq u_j \leq h - 1$), означаващо номера на часа в денонощието, в течение, на който j -ия data-център е недостъпен във връзка с планово обновяване.

От всичко казано до тук следва, че за всеки клиент трябва да се изпълнят условията $u_{c_{i,1}} \neq u_{c_{i,2}}$, тъй като иначе по време на едновременното обновяване на двата data-центъра, компанията няма да бъде в състояние да осигури на този клиент достъп до неговите данни.

Във връзка с различното часово време в различните страни и градове на света, времето за обновяване в някои data-центрове може да се измести с час напред. За подготовка за непредвидени ситуации, ръководството на компанията иска да проведе тренировка, в хода, на която ще бъде избрано някакво непразно подмножество data-центрове, и времето за обновяването на всеки от тях ще бъде изместено с един час по-късно в рамките на денонощието. (т.е., ако $u_j = h - 1$, новият час на обновяване ще бъде 0, иначе новият час на обновяването ще стане $u_j + 1$). При това, тренировката не трябва да наруши гаранцията за достъпност, т.е. след промяна на графика за обновяване, както до този момент трябва да се изпълнява условието, че данните на кой да е клиент ще бъдат достъпни поне в един екземпляр във всеки един час.

Тренировката е полезно, но трудоемко мероприятие и води до големи загуби, затова ръководството на компанията се обръща към вас за помощ при определяне на минимално по размер непразно подходящо подмножество от data-центрове, за да проведе тренировката само в това подмножество.

Input

От първия ред на стандартния вход се въвеждат три цели числа n , m и h ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$, $2 \leq h \leq 100\,000$) — броя на data-центровете на компанията, броя на клиентите на компанията и броя на часовете в денонощието.

На втория ред са ви дадени n числа u_1, u_2, \dots, u_n ($0 \leq u_j < h$), j -тото, от които задава номера на часа, в който се извършва плановете обновяване на програмното осигуряване на компютрите в data-центъра j .

Следват m реда, на всеки, от които има двойка числа $c_{i,1}$ и $c_{i,2}$ ($1 \leq c_{i,1}, c_{i,2} \leq n$, $c_{i,1} \neq c_{i,2}$), задаващи номерата на data-центровете, в които се намират данните на клиента i .

Гарантирано е, че при зададеното разписание на обновяване на data-центровете, за всеки клиент, във всеки момент ще бъде достъпно поне едно копие от неговите данни.

Output

На първия ред изведете минималния брой data-центрове k ($1 \leq k \leq n$), които трябва да засегне тренировката, за да не се загуби гаранцията за достъпност. На втория ред изведете k различни цели числа — номерата на data-центровете x_1, x_2, \dots, x_k ($1 \leq x_i \leq n$), за които в рамките на тренировката обновяванията ще започнат да се провеждат с един час по-късно. Номерата на data-центровете може да се изведат в произволен ред.

Ако възможните отговори са няколко, изведете кой да е от тях. Гарантирано е, че съществува поне един отговор, който удовлетворява условието на задачата.

Examples

input	output
3 3 5 4 4 0 1 3 3 2 3 1	1 3
4 5 4 2 1 0 3 4 3 3 2 1 2 1 4 1 3	4 1 2 3 4

Note

Да разгледаме първия тест от условието. Посоченият отговор е единственият начин да се проведе тренировката така, че да се засегне само един data-център. При такъв сценарий третият сървър започва да се обновява в първия час на деня и никои два сървъра, съхраняващи данни на един и същ потребител не се обновяват в един и същи час.

От друга страна, например, преместването на само на времето за обновяване на първия сървър с един час напред е невъзможно защото в такъв случай данните на потребителите 1 и 3 ще бъдат недостъпни по време на нулевия час.

Scoring

Тестовите към тази задача са разпределени в 3 групи. Точките за всяка група се получават само при преминаване на всички тестове в групата и всички тестове от предходните групи. **Offline-проверка** означава, че резултатите от тестването на вашето решение за дадената група ще станат известни едва след завършване на състезанието.

Група	Точки	Дополнителни ограничения		Коментари
		n	m	
0	0	–	–	Тестовите от условието
1	30	$n \leq 10$	$m \leq 100$	–
2	30	$n \leq 500$	$m \leq 1000$	–
3	40	–	–	Offline-проверка

Problem Michelangelo. Отбой

Input file: input.txt or standard input
Output file: output.txt or standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

Преподавателите на лятната каквато и да е школа, принуждават учениците да си лягат.

Сградата се състои от n стаи, във всяка, от които трябва да живеят точно b ученици. В часа, в който трябва да си лягат, се оказва, че не всички ученици са в стаите си. Стаите са номерирани с числата от 1 до n , като в i -тата стая има a_i ученици. При това, в сградата са всички ученици, които живеят в нея и само те, т.е., $a_1 + a_2 + \dots + a_n = nb$. В сградата също живеят и p преподаватели, като $p \leq 2$.

Процесът на преспиване на учениците, изглежда по следния начин. Един преподавател започва от стая 1 и се движи към стая n . Ако има и втори преподавател, то той започва от n стая и се движи стая 1. След като завърши приспиването в поредната стая, преподавателят отива към следващата. Ако преподавателите са двама, то те влизат и излизат от стаите си едновременно. Ако n е нечетно, а преподавателите двама, то в средната стая приспива само първия преподавател. Когато всички ученици са сложени да спят, процесът завършва.

Когато преподавателят влезе в стая, той преброява учениците, след това гаси осветлението и затваря стаята. Допълнително, ако броят на учениците вътре не съвпада с b , той записва номера на стаята в тефтера си и пак гаси осветлението и затваря стаята. Преподавателите бързат да съставят учебния план за утре, за това не се интересуват кой се намира в стаята, а само от броя на учениците в нея.

Докато преподавателите се намират в стаите, учениците могат да прибегат между още незатворените и неприспани стаи, при това учениците успяват да прибегат през не повече от d стаи (т. е. премества се в стая с номер, който се различава с не повече от d от номера на настоящата). Също така след (или вместо) преместване в друга стая ученик може да се скрие под легло в стаята, в която се намира, тогава преподавателя не го забелязва при преброяването. Във всяка стая под леглата могат да се скрият произволен брой ученици едновременно.

На практика се случва следното:

- Обявява се лягане, в този момент в стая i има a_i ученици.
- Всеки ученик може да се премести в друга стая, но не повече от d стаи спрямо началната или да остане в същата стая. След това желаещите да се крият под леглата се скриват.
- В стая 1 влиза преподавател, а също в стая n , ако преподавателите са двама, слага да спят всички намиращи се там ученици и затваря стаята (след това не трябва нито да се влиза, нито да се излиза от нея).
- Учениците от стаи с номера 2 до n (или до $n - 1$, ако преподавателите са двама) могат да се преместят в други стаи, бягайки, не по далеч от d стаи от своето. **текущо** местоположение, или да останат в същата стая. Желаещите да се скрият под легло, се крият.
- От стая 1 в стая 2 (и възможно е от стая n в стая $n - 1$) преминава преподавател.
- Процесът продължава, докато във всички стаи учениците не бъдат сложени да спят.

Да означим с x_i броя на стаите с нарушения в бройката, които са били записани от i -я преподавател. Учениците знаят, че директорът ще слуша оплаквания за безобразията по време на лягане от не повече от един преподавател, затова искат да действат така, че максималното от числата x_i да е възможно най-малко. Помогнете им да намерят каква минимална стойност може да приема тази величина при оптимални действия на учениците.

Input

На първия ред на стандартния вход са записани четири цели числа p , n , d и b ($1 \leq p \leq 2$, $2 \leq n \leq 100\,000$, $1 \leq d \leq n - 1$, $1 \leq b \leq 10\,000$) — броя на преподавателите, броя на стаите в сградата, максималния брой стаи, които успява да пробяга ученик, докато няма преподавател в коридора, и необходимия брой ученици в една стая.

На втория ред са записани n цели числа a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$), i -то, от които е броят на учениците в i -тата стая преди обявяване на лягането.

Гарантирано е, че $a_1 + a_2 + \dots + a_n = nb$.

Output

Изведете едно число — минималната възможна стойност на максималното x_i .

Examples

input	output
1 5 3 1 0 0 0 5 0	0
1 5 3 10 5 1 1 1 42	1
2 5 1 1 1 0 0 0 4	1
2 6 1 2 3 8 0 1 0 0	2

Note

В първия пример учениците бягат достатъчно бързо, така че да се приберат по стаите си преди преподавател да влезе в първата стая, затова отговорът е 0.

Във втория пример преподавателят ще запише минимум една стая в своя тефтер. Една от оптималните стратегии е следната: преди преподавател да влезе в първата стая, от пета стая по 10 ученици трябва да се преместят в стаи 2, 3 и 4. След това в стаи 2, 3, 4 по един ученик се крие под леглото, в стая 5 под леглото се крият двама ученици, след което учениците не правят нищо повече. При такава последователност от действия преподавателят ще запише в своя тефтер само стая 1.

В третия пример първите три стаи ще посети първият преподавател, а последните две — вторият. Едно от оптималните решения се явява следното: първа стъпка - трима ученици трябва да се преместят от стая 5 в стая 4, втора стъпка - двама от тях трябва да се преместят от стая 3 и един трябва да се скрие под леглото. При това недоволство в първия преподавател ще предизвика само стая 2, а вторият преподавател няма да срещне никакво нарушение.

В четвъртия пример една от оптималните стратегии е следната: първа стъпка - всички ученици от първата стая се скриват, а всички ученици от втората се преместват в третата. Втора стъпка - един ученик се премества от трета в четвърта стая, а още 5 се скриват. По този начин първият преподавател ще е недоволен от стаи 1 и 2, а вторият — от стаи 5 и 6.

Scoring

Тестовите към тази задача са разпределени в осем групи. Точките за всяка група се получават само при преминаване на всички тестове в групата и всички тестове от някои от предходните групи. Обърнете внимание, че преминаването на тестовите от условието **не се изисква** за приемане на решението за проверка. **Offline-проверка** означава, че резултатите от тестването на вашето решение за дадената група ще станат известни едва след завършване на състезанието.

Група	Точки	Дополнителни ограничения			Необх. групи	Коментари
		p	n	b		
0	0	–	–	–	–	Тестовете от условието
1	10	$p = 1$	$n \leq 1000$	$b = 1$	–	–
2	10	$p = 1$	$n \leq 1000$	–	1	–
3	10	–	$n \leq 100$	$b = 1$	–	–
4	15	–	$n \leq 100$	$b \leq 30$	0, 3	–
5	20	–	$n \leq 1000$	$b \leq 30$	0, 1, 3, 4	–
6	10	$p = 1$	–	–	1, 2	Offline-проверка
7	25	–	–	–	0–6	Offline-проверка

Problem Munch. Двоични карти

Input file: input.txt or standard input
Output file: output.txt or standard output
Time limit: 1 second
Memory limit: 512 megabytes

Никога не е късно човек да се научи да играе на увлекателната игра "Двоични карти"!

В нея участват неограничен брой игрални карти с различни положителни и отрицателни стойности. Абсолютната величина, записана на всяка карта е степен на двойката, т. е. на картата може да бъде написано или 2^k , или -2^k за някое цяло число $k \geq 0$. Има неограничен брой карти от всяка стойност.

В началото играчът избира за себе си колода — някакво множество (възможно празно) игрални карти. При това в колодата е разрешено да бъдат включени произволен брой карти от всяка стойност, но нивото на играча се оценява по това, дали той включва в своята колода минимално възможен брой карти. Играта се състои от n рунда. На i -тия рунд журито задава на играча цяло число a_i . След нова играчът трябва да постави на масата такова подмножество карти от своята колода, че сборът от стойностите на тези карти е точно равен на a_i (може да не се поставят на масата никакви карти и тогава сборът се приема за равен на нула). Ако играчът не може да постави на масата нужния брой карти, се приема, че той е загубил и играта завършва. В противен случай играчът връща извадените карти в своята колода и преминава към следващият рунд. Играчът печели когато във всеки рунд е успял да извади необходимите карти.

Вие сте успели да съберете информация за това какви числа a_i ще назовава журито на всеки рунд. Сега на вас ви остава само да изберете колода с минимален брой карти, която ви позволява да спечелите играта "Двоични карти".

Input

На първия ред на стандартния вход е записано цяло число n ($1 \leq n \leq 100\,000$) — броя числа в играта.

На втория ред са записани n цели числа a_1, a_2, \dots, a_n ($-100\,000 \leq a_i \leq 100\,000$) — числата, които журито ще задава на всеки рунд.

Output

На първия ред на стандартния изход изведете едно число k ($0 \leq k \leq 100\,000$) — минималния брой карти, които трябва да изберете в колодата, така че да спечелите в играта "Двоични карти".

На втория ред изведете k цели числа b_1, b_2, \dots, b_k ($-2^{20} \leq b_i \leq 2^{20}$, $|b_i|$ са степени на двойката) — стойностите на картите, включени в колодата. Стойностите може да се изведат в произволен ред. Ако съществуват няколко колоди с оптимален размер, изведете коя да е от тях.

Гарантирано е, че съществува колода от карти с минимален размер, удовлетворяваща изискванията на стойностите на числата, представени по-горе.

Examples

input	output
1 9	2 1 8
5 -1 3 0 4 7	3 4 -1 4
4 2 -2 14 18	3 -2 2 16

Note

В първия тест от условието при единственият рунд се изваждат и двете карти от колодата. Обърнете внимание, че този тест от условието — е единствен от подходящите при ограниченията на първата група тестове.

Във втория тест от условието на първия рунд може да се извади единствена карта -1 , във втория рунд — картите 4 и -1 , в третия рунд — нищо, в четвъртия — само картата 4 , а в петия рунд — цялата колода.

Scoring

Тестовите към тази задача са разпределени в 14 групи. Точките за всяка група се получават само при преминаване на всички тестове в групата и всички тестове от предходните групи. Обърнете внимание, че преминаването на тестовите от условието **не се изисква** за приемане на решението за проверка. **Offline-проверка** означава, че резултатите от тестването на вашето решение за дадената група ще станат известни едва след завършване на състезанието.

Група	Точки	Допълнителни ограничения		Коментари
		n	$ a_i $	
0	0	–	–	Тестовите от условието
1	8	$n = 1$	$ a_i \leq 10$	–
2	7	$n \leq 10$	$ a_i \leq 10$	–
3	7	$n \leq 30$	$ a_i \leq 30$	–
4	7	$n \leq 50$	$ a_i \leq 50$	–
5	7	$n \leq 100$	$ a_i \leq 100$	–
6	7	$n \leq 300$	$ a_i \leq 300$	–
7	8	$n \leq 500$	$ a_i \leq 500$	–
8	7	$n \leq 1000$	$ a_i \leq 1000$	Offline-проверка
9	6	$n \leq 3000$	$ a_i \leq 3000$	Offline-проверка
10	7	$n \leq 5000$	$ a_i \leq 5000$	Offline-проверка
11	6	$n \leq 10\,000$	$ a_i \leq 10\,000$	Offline-проверка
12	7	$n \leq 30\,000$	$ a_i \leq 30\,000$	Offline-проверка
13	7	$n \leq 50\,000$	$ a_i \leq 50\,000$	Offline-проверка
14	9	–	–	Offline-проверка