

## Задача А. Загадочный лес 1

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ёжик гулял по загадочному лесу и встретил таинственного деда волшебника. Он предложил ежику пройти пять испытаний. Если он справится, то получит подарок. Ёжик конечно не стал отказываться от такого предложения. Сейчас он проходит первое по сложности (самое простое) испытание.

Волшебник очень таинственный, поэтому он дал Ёжику следующее задание: написать реализацию функции:

```
int solve(int a, int b, int c) {  
    // ...  
}
```

Конечно не любую функцию Ёжик может написать и она должна выводить такой же результат, как и [https://ugadalka.algocourses.ru/?task\\_id=1](https://ugadalka.algocourses.ru/?task_id=1). Ограничения на входные данные:  $1 \leq a, b, c \leq 1000$ .

Гарантируется, что существует реализация функции без ветвлений (условного оператора, тернарного оператора и т.д.). Также Дед очень честный и не использовал в функции какие-то странные числа (например 1231, 8190, 52, ...).

Отправьте реализацию функцию `solve` в следующем формате:

```
int solve(int a, int b, int c) {  
    // YOUR CODE  
}
```

Ничего кроме функции `solve` сдавать **не нужно**.

## Задача В. Загадочный лес 2

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ёжик гулял по загадочному лесу и встретил таинственного деда волшебника. Он предложил ежику пройти пять испытаний. Если он справится, то получит подарок. Ёжик конечно не стал отказываться от такого предложения. Сейчас он проходит второе по сложности испытание.

Волшебник очень таинственный, поэтому он дал Ёжику следующее задание: написать реализацию функции:

```
int solve(int a, int b, int c) {  
    // ...  
}
```

Конечно не любую функцию Ёжик может написать и она должна выводить такой же результат, как и [https://ugadalka.algocourses.ru/?task\\_id=2](https://ugadalka.algocourses.ru/?task_id=2). Ограничения на входные данные:  $1 \leq a, b, c \leq 1000$ .

Гарантируется, что существует реализация функции без ветвлений (условного оператора, тернарного оператора и т.д.). Также Дед очень честный и не использовал в функции какие-то странные числа (например 1231, 8190, 52, ...).

Отправьте реализацию функцию `solve` в следующем формате:

```
int solve(int a, int b, int c) {  
    // YOUR CODE  
}
```

Ничего кроме функции `solve` сдавать **не нужно**.

## Задача С. Загадочный лес 3

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ёжик гулял по загадочному лесу и встретил таинственного деда волшебника. Он предложил ежику пройти пять испытаний. Если он справится, то получит подарок. Ёжик конечно не стал отказываться от такого предложения. Сейчас он проходит третье по сложности испытание.

Волшебник очень таинственный, поэтому он дал Ёжику следующее задание: написать реализацию функции:

```
int solve(int a, int b, int c) {  
    // ...  
}
```

Конечно не любую функцию Ёжик может написать и она должна выводить такой же результат, как и [https://ugadalka.algocourses.ru/?task\\_id=3](https://ugadalka.algocourses.ru/?task_id=3). Ограничения на входные данные:  $1 \leq a, b, c \leq 1000$ .

Гарантируется, что существует реализация функции без ветвлений (условного оператора, тернарного оператора и т.д.). Также Дед очень честный и не использовал в функции какие-то странные числа (например 1231, 8190, 52, ...).

Отправьте реализацию функцию `solve` в следующем формате:

```
int solve(int a, int b, int c) {  
    // YOUR CODE  
}
```

Ничего кроме функции `solve` сдавать **не нужно**.

## Задача D. Загадочный лес 4

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ёжик гулял по загадочному лесу и встретил таинственного деда волшебника. Он предложил ежику пройти пять испытаний. Если он справится, то получит подарок. Ёжик конечно не стал отказываться от такого предложения. Сейчас он проходит четвертое по сложности испытание.

Волшебник очень таинственный, поэтому он дал Ёжику следующее задание: написать реализацию функции:

```
int solve(int a, int b, int c) {  
    // ...  
}
```

Конечно не любую функцию Ёжик может написать и она должна выводить такой же результат, как и [https://ugadalka.algocourses.ru/?task\\_id=4](https://ugadalka.algocourses.ru/?task_id=4). Ограничения на входные данные:  $1 \leq a, b, c \leq 1000$ .

Гарантируется, что существует реализация функции без ветвлений (условного оператора, тернарного оператора и т.д.). Также Дед очень честный и не использовал в функции какие-то странные числа (например 1231, 8190, 52, ...).

Отправьте реализацию функцию `solve` в следующем формате:

```
int solve(int a, int b, int c) {  
    // YOUR CODE  
}
```

Ничего кроме функции `solve` сдавать **не нужно**.

## Задача Е. Загадочный лес 5

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ёжик гулял по загадочному лесу и встретил таинственного деда волшебника. Он предложил ежику пройти пять испытаний. Если он справится, то получит подарок. Ёжик конечно не стал отказываться от такого предложения. Сейчас он проходит пятое по сложности (самое сложное) испытание.

Волшебник очень таинственный, поэтому он дал Ёжику следующее задание: написать реализацию функции:

```
int solve(int a, int b, int c) {  
    // ...  
}
```

Конечно не любую функцию Ёжик может написать и она должна выводить такой же результат, как и [https://ugadalka.algocourses.ru/?task\\_id=5](https://ugadalka.algocourses.ru/?task_id=5). Ограничения на входные данные:  $1 \leq a, b, c \leq 1000$ .

Гарантируется, что существует реализация функции без ветвлений (условного оператора, тернарного оператора и т.д.). Также Дед очень честный и не использовал в функции какие-то странные числа (например 1231, 8190, 52, ...).

Отправьте реализацию функцию `solve` в следующем формате:

```
int solve(int a, int b, int c) {  
    // YOUR CODE  
}
```

Ничего кроме функции `solve` сдавать **не нужно**.

## Задача F. Пингвины и скалы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пингвины собираются устроить свою собственную олимпиаду. Самый главный и единственный спорт, включенный в эту олимпиаду это прыжки по скалам. Но согласно правилам данного общеизвестного спорта, высоты скал, записанные слева направо должны составлять арифметическую прогрессию.

Организаторы подготовили для прыжков линию из  $n$  скал с высотами  $a_1, a_2, \dots, a_n$ , но совершенно забыли про правило с арифметической прогрессией, и теперь при подаче документов на согласование надо вычеркнуть из плана не более  $k$  скал (иначе кто-то может что-то заподозрить), чтобы оставшиеся оказались какой-то арифметической прогрессией. При этом в целях конспирации организаторы хотят удалить минимальное количество скал. Помогите им пожалуйста.

### Формат входных данных

Первая строка входных данных содержит числа  $n, k$  ( $2 \leq n \leq 2 \cdot 10^5, 0 \leq k \leq \min(n, 500)$ ) — количество скал и сколько скал можно вычеркнуть.

Вторая строка входных данных содержит  $n$  чисел  $a_1, a_2, \dots, a_n$  ( $-10^6 \leq a_i \leq 10^6$ ) — высоты скал.

### Формат выходных данных

Если организаторы не смогут вычеркнуть скалы требуемым образом, выведите  $-1$ .

Иначе в первой строке выведите минимальное число вычёркиваний, которое надо сделать, а во второй номера вычёркнутых скал.

### Пример

стандартный ввод	стандартный вывод
7 2	2
2 4 6 7 8 9 10	4 6

## Задача G. Волшебный мешок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В далеком снежном королевстве жил великий волшебник — Дед Мороз. Каждый год он создавал особый Волшебный Мешок, чтобы доставить подарки детям по всему миру. В мешке были числа, каждое из которых символизировало силу новогоднего чуда. Чем больше это число, тем ярче светятся елки, громче смех детей и сильнее дух праздника.

Однако волшебство этих чисел подчинялось древнему закону числовой магии. Когда Дед Мороз складывал числа в своем мешке, их суммарная энергия должна была усиливаться на каждом этапе. Это измерялось величиной, которую древние математики называли наибольшим общим делителем (GCD). Чем больше этот GCD, тем мощнее становилась магия префиксных сумм — это значит, что каждое промежуточное сложение давало больше праздничной силы.

Но вот незадача: числа в мешке Деда Мороза оказались перепутаны! Чтобы подготовить подарки вовремя, Деду Морозу нужно найти такой порядок чисел, чтобы максимизировать магическую силу GCD их префиксных сумм.

### Формат входных данных

В первой строке вводится целое число  $n$  — количество подарков ( $1 \leq n \leq 10^5$ ).

Во второй строке вводятся  $n$  целых чисел — магические силы подарков ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

В первой строке выведите максимальный возможный GCD.

Во второй строке выведите перестановку длины  $n$  — оптимальный порядок подарков.

### Примеры

стандартный ввод	стандартный вывод
3 2 6 4	2 2 1 3
4 5 1 6 9	1 4 3 2 1

## Задача Н. Взлом ПИН-кода

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

### Это интерактивная задача.

ПИН-код для банковской карты задаётся строкой из четырёх цифр. Возможные значения ПИН-кода от «0000» до «9999».

Внимательно ознакомьтесь с документом <https://www.ietf.org/rfc/rfc1321.html>. В нём описан алгоритм вычисления md5 хэша.

Из соображений безопасности ПИН-коды VIP клиентов не хранятся в базе данных явно, вместо этого банк хранит их md5 хэши.

Вашей группировке удалось похитить один такой md5 хэш. Подберите подходящий ПИН-код. У вас будет **три** попытки до того, как карта окажется заблокирована.

### Протокол взаимодействия

Вы начинаете взаимодействие со считывания строки длины 32 — валидного md5 хэша загаданного ПИН-кода.

Вы можете не более **трёх** раз ввести на проверку ПИН-код.

Если ПИН-код **верный**, интерактор выведет «ok». После получения такого сообщения вы обязаны немедленно завершить программу.

Если ПИН-код **неверный**, интерактор выведет «wrong». После получения такого сообщения вы можете продолжать проверять ПИН-коды, если у вас ещё остались попытки.

### Пример

стандартный ввод	стандартный вывод
312351bff07989769097660a56395065	
wrong	1234
wrong	0000
ok	2025



## Задача I. Снегурочка и ор

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 1024 мегабайта

Снегурочка пыталась понять, как работает ор (побитовый, обычный она понимает, как работает).  $2^n$  лесных животных предложили ей простую задачу. Они выстроились в ряд, высота  $i$ -го животного равен  $a_i$ . Нумерация животных с нуля. И предложили ей для каждого  $1 \leq k \leq 2^n - 1$  найти максимальную сумму  $a_i + a_j$  по всем парам животных таким, что  $0 \leq i < j \leq 2^n - 1$  и  $i|j \leq k$ , где  $|$  обозначает операцию побитового или.

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 18$ )

Следующая строка содержит  $2^n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ )

### Формат выходных данных

Выведите в  $2^n - 1$  строках ответ для каждого  $k$ .

### Примеры

стандартный ввод	стандартный вывод
2 1 2 3 1	3 4 5
3 10 71 84 33 6 47 23 25	81 94 155 155 155 155 155
4 75 26 45 72 81 47 97 97 2 2 25 82 84 17 56 32	101 120 147 156 156 178 194 194 194 194 194 194 194 194 194 194

## Задача J. Новогоднее дерево

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Маленький олененок гулял по лесу и нашел граф. Всего в графе  $n$  вершин, пронумерованные от 1 до  $n$ . Между каждой парой вершин есть ребро. Из данного графа олененок решил сделать дерево. Чтобы оно было больше похоже на новогоднее, он решил построить на графе минимальное остовное дерево, где вес каждого ребра побитовое **and** чисел вершин ребра.

Побитовое **and** — бинарная операция, действие которой эквивалентно применению логического «И» к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Например, побитовое **and** 13 и 6 = 4. Данная операция существует во всех современных языках программирования, например, в языках C++, Java и Python она обозначена как «&», в Pascal — как «and».

### Формат входных данных

В первой строке вводится целое число  $n$  — количество вершин в графе ( $1 \leq n \leq 10^5$ ).

### Формат выходных данных

В единственной строке входных данных выведите вес минимального остовного дерева данного графа.

### Примеры

стандартный ввод	стандартный вывод
2	0
4	0

## Задача К. Крестик из sudoku: сумма

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Решите усложнённое sudoku. В каждом квадрате  $9 \times 9$  должны выполняться правила sudoku: в каждой горизонтали, каждой вертикали и каждом выделенном квадрате  $3 \times 3$  должны присутствовать все цифры от 1 до 9.

			4	5		3													2	
					1				8			9					7	5	1	
5	9								4	7						9			3	
	7	8																	4	8
4	6		9		8															
1														3						
				3	4	2		9											2	
						8	5			1						5			1	
	5	3			2				2							7		4		
						8			4	2										
								1	3		8									
			9	6					9				1					4	8	
		1		8					4			9	6							
		2									1	4	5	7						
					2															8
																			1	
	6	3				2	9												8	5
	8			5												4			2	7
	9	7	5						3										1	9
3				4		8														
										1		2	9							

### Формат выходных данных

Чтобы доказать, что вам удалось решить это sudoku, сдайте программу, которая выводит только одно число: сумму всех цифр, которые вы поставили в пустые клетки в правильном решении.

## Задача L. Крестик из sudoku: диагональ

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Решите усложнённое sudoku. В каждом квадрате  $9 \times 9$  должны выполняться правила sudoku: в каждой горизонтали, каждой вертикали и каждом выделенном квадрате  $3 \times 3$  должны присутствовать все цифры от 1 до 9.

			4	5		3												3		7										2						
							1							8						9						7	5	1								
5	9												4	7							9							3								
	7	8												5							7	2				4	8									
4	6			9			8														8															
1																							3													
					3	4	2			9																	2									
								8	5				1									5					1									
		5	3				2						2									7	4													
								8			4		2																							
												1		3			8																			
					9	6										9						1				4	8									
			1			8						4					9	6																		
			2														1	4	5	7																
								2																												
														1																						
		6	3																			8														
		8				5																4														
		9	7	5																								2	7							
		3																				3	2													
						4		8																												

### Формат выходных данных

Чтобы точно доказать, что вам удалось решить это sudoku, сдайте программу, которая выводит только одну строку из 21 цифры: содержимое главной диагонали в правильном решении без пробелов (от левого верхнего угла к правому нижнему).

## Задача М. Хорошие раскраски New Year edition

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 секунд
Ограничение по памяти:	256 мегабайт

Эта задача даже не очень задача. От вас лишь требуется раскрасить картинки, которые мы вам даем в архиве. В нем есть некоторое количество примеров, как должны выглядеть раскрашенные и нераскрашенные изображения. Также в архиве лежат все тесты, на которых будут тестироваться ваши решения. Для вашего удобства в архиве также лежит файл «base.cpp», в котором показан пример того, как должен производиться ввод и вывод в этой задаче.

<http://ejudge.lksh.ru/olymp/paintings.zip>

### Формат входных данных

Во входных данных вы получаете файл в формате ppm. Для вас уже был написан код, который вводит файл, сохраняет его в табличку и выводит его.

### Формат выходных данных

В выходных данных выведете картинку в формате ppm.

### Примеры

стандартный ввод	стандартный вывод
P3 654 736 255 ...	P3 654 736 255 ...
P3 677 717 255 ...	P3 677 717 255 ...

### Замечание

В первом семпле отображается лишь начало файла «bird.ppm» так как он слишком большой.  
Во втором семпле отображается лишь начало файла «totoro.ppm» так как он слишком большой.

## Задача N. Сортировка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дана перестановка  $p$  длины  $n$  и  $m$  пар чисел  $x_i, y_i$  от 1 до  $n$ . Требуется обработать  $q$  запросов, каждый запрос это пара чисел  $1 \leq a, b \leq n$ , при запросе нужно поменять  $p_a$  и  $p_b$  местами. Требуется определить, можно ли отсортировать перестановку какой-то последовательностью обменов элементов с индексами  $x_i, y_i$  после каждого запроса.

### Формат входных данных

В первой строке вводятся три числа —  $n, m, q$ . Гарантируется, что  $1 \leq n, m, q \leq 2 \cdot 10^5$ .

Во второй строке содержатся  $n$  чисел —  $p_1, p_2, \dots, p_n$ . Гарантируется, что  $p$  — перестановка длины  $n$ .

Следующие  $m$  строк содержат пары  $x_i, y_i$ . Гарантируется, что  $1 \leq x_i, y_i \leq n$ . Следующие  $q$  строк содержат запросы, то есть пары  $a_i, b_i$ . Гарантируется, что  $1 \leq a_i, b_i \leq n$ .

### Формат выходных данных

Введите строку длины  $q$  из символов «N» и «Y».

## Примеры

стандартный ввод	стандартный вывод
5 3 6 1 4 2 3 5 3 4 1 2 4 1 1 3 1 4 3 4 3 1 3 4 5 5	YYYYYY
5 1 7 1 3 2 4 5 2 3 2 4 1 2 2 3 3 4 4 1 1 2 2 3	NNNNYNN
7 7 10 3 2 1 5 4 6 7 3 1 3 4 2 4 1 4 2 5 5 5 1 2 7 1 4 6 4 6 1 7 2 2 3 7 3 7 5 4 2 6 2 6	NNNYNYNY

## Задача О. Странная гирлянда

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Зимой в ЛКШ стояла огромная красивая елка, все ученики часто приходили на нее смотреть, зачем? Потому что на этой елке вместо гирлянды был массив целых чисел длины  $n$ , ученики заглядывались на все подотрезки **четной** длины и считали их красоту в виде целого числа которое определялось следующим образом:

- Для массива **четной** длины определим его красоту как наибольший общий делитель всех чисел, которые являются суммой образованной парой чисел равноудаленных от его середины.

Например красота гирлянды  $(16, 11, 20, 24)$  будет равна 1, так как  $16 + 24 = 40, 11 + 20 = 31$ ,  $\text{НОД}(\{31, 40\}) = 1$ .

Вам стало интересно насколько красивой благодаря гирлянде стала вся елка, красоту елки определим как сумму красот для всех подотрезков массива на гирлянде четной длины, найдите это число.

### Формат входных данных

Первая строка содержит единственное число  $n$  — длина гирлянды ( $2 \leq n \leq 100000$ ).

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  — числа на гирлянде ( $1 \leq a_i \leq 100$ ).

### Формат выходных данных

Выведите одно число — красоту елки.

### Пример

стандартный ввод	стандартный вывод
5 1 2 3 4 5	36

### Замечание

В примере имеем 4 подотрезка длины 2, их красоты равны 3, 5, 7, 9 слева направо.

Также мы имеем 2 подотрезка длины 4, красоты равны 5 и 7 соответственно.

Тогда ответ  $(3 + 5 + 7 + 9) + (5 + 7) = 36$ .



## Задача Р. Время (TIME)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

<https://timelksh.algocourses.ru/>

## Задача Q.

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 1024 мегабайта

algotup тулзы : скальп ,  
бобан ,  
фара , шогa , гибон , бланка  
, выдра , каспер , валерия , пепел , вуда , барон  
, рипер ( не Джек )  
, ёжик ,  
постик  
.

## Задача R. Гномик и таблички

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Гномик Нубик-Дровосек нашел две таблички размера  $n \times m$ . Каждая ячейка каждой таблички покрашена в какой-то цвет, причем гномик заметил, что для любой картины выполнено следующее:

- Какой бы столбец он ни выбрал, все цвета в нем будут уникальны
- Какую бы строку он ни выбрал, все цвета в ней будут уникальны

Исключением являлся цвет 0: он мог повторяться как в строках, так и в столбцах.

Нубик-Дровосек хочет получить из таблички  $a$  табличку  $b$  при помощи следующих изменений:

- Выбрать либо две строки, либо два столбца таблички и поменять их местами.

Помогите гномику понять, сможет ли он достичь желаемого (и если сможет, то приведите нужную последовательность обменов, ведь без вас Нубику не справиться!)

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $m$  ( $1 \leq n \cdot m \leq 2 \cdot 10^5$ ) — размеры табличек.

В  $i$ -й из следующих  $n$  строк записаны  $m$  целых чисел  $a_{i,1}, a_{i,2}, \dots, a_{i,m}$  ( $0 \leq a_{i,j} \leq 2 \cdot 10^5$ ) — цвета в  $i$ -й строке таблички  $a$ . Нумерация строк — от 1 до  $n$  сверху вниз, столбцов — от 1 до  $m$  слева направо. Гарантируется, что табличка удовлетворяет ограничениям, описанным в условии задачи.

В  $i$ -й из следующих  $n$  строк записаны  $m$  целых чисел  $b_{i,1}, b_{i,2}, \dots, b_{i,m}$  ( $0 \leq b_{i,j} \leq 2 \cdot 10^5$ ) — цвета в  $i$ -й строке рисунка  $b$ . Нумерация строк такая же. Гарантируется, что табличка удовлетворяет ограничениям, описанным в условии задачи.

### Формат выходных данных

В первой строке выведите число  $-1$ , если добиться желаемого невозможно, иначе выведите число действий в вашем решении  $k$  ( $0 \leq k \leq 2 \cdot 10^5$ ). Можно доказать, что если решение существует, то существует решение, где  $k \leq 2 \cdot 10^5$ .

В следующих  $k$  строках выведите ваши действия. Сначала выведите тип изменения (1 — поменять местами строки, 2 — столбцы), а потом два номера (строк или столбцов), к которым применяется операция.

Обратите внимание, что вам минимизировать число операций не требуется.

## Примеры

стандартный ввод	стандартный вывод
3 3 1 0 2 0 0 0 2 0 1 2 0 1 0 0 0 1 0 2	1 1 1 3
4 4 0 0 1 2 3 0 0 0 0 1 0 0 1 0 0 0 2 0 0 1 0 3 0 0 0 1 0 0 0 0 1 0	4 1 3 4 2 3 4 2 2 3 2 1 2
3 3 1 2 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 0	-1

## Задача S. Копим деньги

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дед мороз копит деньги на подарки всем к новому году, и поскольку он очень ленив деньги ему приносят только эльфы, живущие у него дома. К моменту нового года у деда мороза дома будут жить  $n$  эльфов,  $i$ -й эльф приходит в дом в момент времени  $T_i$  и приносит 1 монету каждые  $R_i$  секунды. За какое минимальное кол-во секунд дед мороз сможет накопить  $X$  монет для того, чтобы купить подарки.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n, X$  ( $1 \leq n \leq 10^5, 1 \leq X \leq 10^9$ ) — кол-во эльфов, сколько монет требуется.

Следующие  $n$  строк содержат два целых числа  $T_i, R_i$  ( $1 \leq T_i \leq 10^9, 1 \leq R_i \leq 10^3$ ) — время когда эльф придет и его период.

### Формат выходных данных

Выведите ответ на задачу))

### Пример

стандартный ввод	стандартный вывод
2 13 5 4 3 2	21

### Замечание

Ко времени 21 первый эльф принесет 4 монеты, второй эльф принесет 9 монеты.

Ко времени 11 первый эльф принесет 1 монету, второй эльф принесет 4 монеты, суммарно будет 5 монет, поэтому ответ не равен 11.

## Задача Т. Новогоднее дерево

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Ноко Шиканoko приготовилась наряжать новогоднее дерево. Напомним, что новогодним деревом является связанный граф на  $n$  вершинах с  $n - 1$  ребром. Ноко Шиканoko называет шириной корневого дерева максимальное число вершин с одинаковой глубиной до корня. Красотой дерева Ноко Шиканoko называет максимальную ширину по всем возможным выборам корня дерева.

Для  $t$  тестов Ноко Шиканoko стало интересно, какой красотой обладает дерево на  $n$  вершинах, если для любого  $i \in [1, n)$  существует ребро между вершинами  $i$  и  $n \bmod i$ . Заметьте, что вершины имеют 0-нумерацию. Можно показать, что граф образованный такими ребрами действительно является деревом.

### Формат входных данных

В первой строке входных данных содержится целое число  $t$  ( $1 \leq t \leq 2 \cdot 10^5$ ) — число деревьев, для которых Ноко Шиканoko хочет узнать их красоту.

В следующих  $t$  строках содержится по одному целому числу  $n_i$  ( $1 \leq n_i \leq 2 \cdot 10^5$ ) — размеры деревьев.

Гарантируется, что  $\sum_{i=1}^t n_i \leq 2 \cdot 10^5$ .

### Формат выходных данных

Выведите  $t$  строк — красоты деревьев.

### Пример

стандартный ввод	стандартный вывод
5	2
3	4
7	4
9	7
15	15
43	

## Задача U. Необычное знакомство

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Саше понравилась девочка. К сожалению, между Сашей и девочкой встала огромная грозная горилла, поэтому он боится подойти и познакомиться. Единственным вариантом остаётся как-то узнать телеграмм этой девочки, чтобы позже связаться с ней.

Саша очень торопится прочитать лекцию, поэтому у него нет времени решать эту задачу. Напишите код, который будет вставлен в начало следующего фрагмента:

```
// YOUR code

class Gorilla {
    // some secret code
};

class Tyan {
    // some secret code
};

const char* get_tg(const Tyan& tyan) {
    return "Meow!";
}
```

Помогите Саше добиться того, чтобы функция `get_tg` возвращала телеграмм переданной девочки.

### Замечание

- Подпрограмма `main`, вызывающая функцию `get_tg`, находится в **другой** единице трансляции.
- Ваше решение **не** должно что-то считывать или что-то выводить.