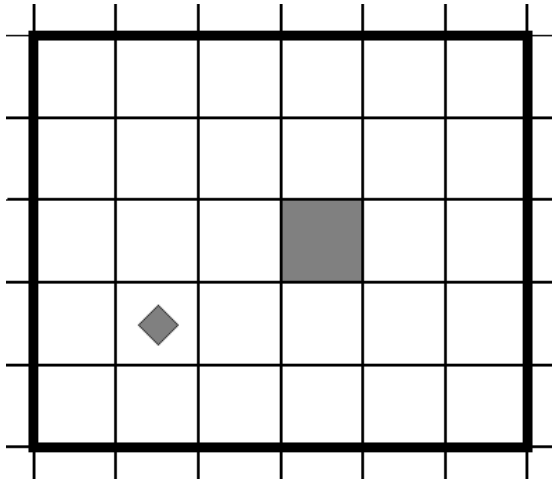


## Задача 4. Робот и клетка

Робот находится на прямоугольном поле в какой-то из клеток. Известно, что на поле есть ровно одна закрашенная клетка.



Пример возможного теста, совпадает с первым тестом жюри.

Перед вами стоит задача переместить Робота в закрашенную клетку. Гарантируется, что количество строк и количество столбцов поля не менее четырёх и не более двадцати. Допустимо, что изначально Робот уже находится в закрашенной клетке.

### Формат входных данных

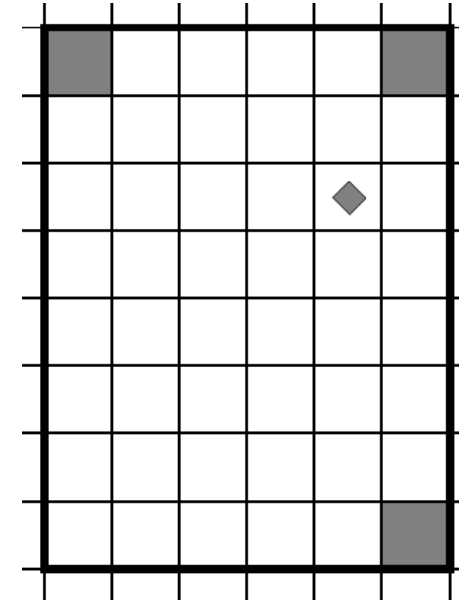
Задачи на Робота не предполагают чтение каких-либо входных данных. Обратите внимание, что жюри в качестве поля может выбрать любое возможное поле, которое соответствует описанию выше.

### Формат выходных данных

Задачи на Робота не предполагают вывод каких-либо данных. Для определения правильности работы вашей программы, будет проанализировано поведение Робота.

## Задача 5. Робот и углы

Робот находится на прямоугольном поле в какой-то из клеток. Известно, что на поле есть ровно три закрашенные клетки, причем все они расположены в угловых клетках поля. Таким образом, на поле есть ровно одна угловая клетка, которая не закрашена.



Пример возможного теста, совпадает с первым тестом жюри.

Перед вами стоит задача переместить Робота в угловую клетку поля, которая не закрашена (такая клетка на поле ровно одна). Гарантируется, что количество строк и количество столбцов поля не менее четырёх и не более двадцати.

### Формат входных данных

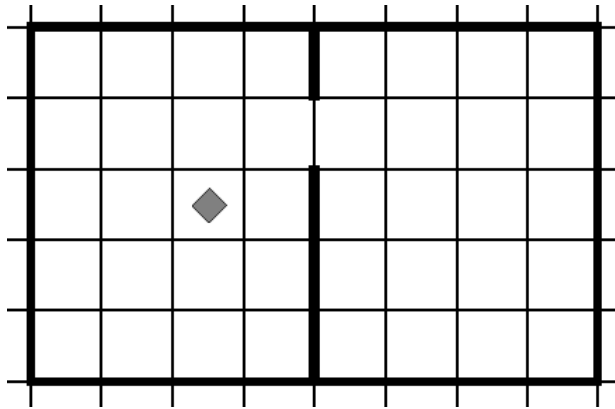
Задачи на Робота не предполагают чтение каких-либо входных данных. Обратите внимание, что жюри в качестве поля может выбрать любое возможное поле, которое соответствует описанию выше.

### Формат выходных данных

Задачи на Робота не предполагают вывод каких-либо данных. Для определения правильности работы вашей программы, будет проанализировано поведение Робота.

## Задача 6. Робот и команды

Робот находится на прямоугольном поле в какой-то из клеток. Известно, что на поле есть ровно одна большая вертикальная стена, расположенная между двумя **произвольными** столбцами клеток на поле. Эта стена проходит с самого верха до самого низа, при этом в этой стене есть ровно один проход, длина которого равна одной клетке. Эта стена разбивает поле на две комнаты — на левую и на правую. Гарантируется, что проход стены не граничит с верхней и нижней границами поля, другими словами стена обязательно примыкает к верхней и нижней границам поля.



Пример возможного теста, совпадает с первым тестом жюри.

Перед вами стоит задача переместить Робота в правую верхнюю клетку поля, если изначально он находится в левой комнате. В противном случае, если Робот изначально находится в правой комнате, вам нужно переместить Робота в левую нижнюю клетку поля. Гарантируется, что количество строк и количество столбцов поля не менее четырёх и не более двадцати.

#### Формат входных данных

Задачи на Робота не предполагают чтение каких-либо входных данных. Обратите внимание, что жюри в качестве поля может выбрать любое возможное поле, которое соответствует описанию выше.

#### Формат выходных данных

Задачи на Робота не предполагают вывод каких-либо данных. Для определения правильности работы вашей программы, будет проанализировано поведение Робота.

## Задача 7. Компот

У Николая есть  $a$  лимонов,  $b$  яблок и  $c$  груш. Он решил приготовить из них компот. По рецепту фрукты должны быть в соотношении  $1 : 2 : 4$ , то есть на каждый лимон в компоте должно быть ровно 2 яблока и ровно 4 груши. При этом фрукты нельзя измельчать, ломать или разрезать, поэтому в компот обязательно должны попасть целиком фрукты — лимоны, яблоки и груши.

Перед вами стоит задача определить максимальное суммарное количество лимонов, яблок и груш, из которых можно сварить компот по рецепту. Возможна ситуация, когда Николай не сможет для приготовления компота использовать ни одного фрукта, в таком случае выведите 0.

#### Формат входных данных

В первой строке следует целое положительное число  $a$  ( $1 \leq a \leq 1000$ ) — количество лимонов, которые есть у Николая

Во второй строке следует целое положительное число  $b$  ( $1 \leq b \leq 1000$ ) — количество яблок, которые есть у Николая.

В третьей строке следует целое положительное число  $c$  ( $1 \leq c \leq 1000$ ) — количество груш, которые есть у Николая.

#### Формат выходных данных

Выведите максимальное суммарное количество лимонов, яблок и груш, из которых можно сварить компот по рецепту.

#### Примеры

стандартный ввод	стандартный вывод
2 5 7	7
4 7 13	21
2 3 2	0

#### Замечание

В первом примере Николай может использовать для приготовления компота 1 лимон, 2 яблока и 4 груши, поэтому ответ  $1 + 2 + 4 = 7$ .

Во втором примере Николай может использовать для приготовления компота 3 лимона, 6 яблок и 12 груш, поэтому ответ  $3 + 6 + 12 = 21$ .

В третьей примере Николаю не хватит груш, чтобы приготовить сколько-нибудь компота, поэтому ответ 0.

## Задача 8. Конфеты на Новый год

У Деда Мороза есть  $n$  конфет, которые он хочет поскорее раздать детям на Новый год.

Перед вами стоит задача определить максимальное число детей, которые получают конфеты от Деда Мороза, если он хочет, чтобы каждый из детей, которым он даст конфеты, получил **различное** целое положительное количество конфет. Дед Мороз хочет раздать все  $n$  конфет, которые у него есть.

#### Формат входных данных

В единственной строке записано целое положительное число  $n$  ( $1 \leq n \leq 1000$ ) — количество конфет у Деда Мороза.

#### Формат выходных данных

В первую строку выведите целое число  $k$  — максимальное число детей, которые получают конфеты.

Во второй строке выведите  $k$  **различных** целых чисел — сколько конфет получит каждый из  $k$  детей. Сумма выведенных  $k$  чисел должна быть равна  $n$ , так как Дед Мороз хочет раздать все конфеты.

Если решений несколько, выведите любое.

#### Примеры

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

## Задача 9. Расшифровка

Поликарп без ума от шифрования, поэтому он пишет Свете сообщения в зашифрованном виде. *Медианной буквой* в слове он называет ту, которая находится в середине слова. Если слово четной длины, то медианная буква та, которая стоит левее из двух средних букв. Медианная буква выделена в следующих примерах: **contest**, **info**. Если слово состоит из одной буквы, то, по определению выше, она и будет его медианной буквой.

Поликарп шифрует каждое слово следующим образом: он выписывает медианную букву слова, затем удаляет эту букву из слова и повторяет процесс до тех пор, пока в слове есть хоть одна буква. Например, слово **volga** он зашифрует в слово **logva**.

Вам задано слово  $s$ , зашифрованное Поликарпом, а ваша задача — расшифровать его.

#### Формат входных данных

В первой строке следует целое положительное число  $n$  ( $1 \leq n \leq 2000$ ) — длина зашифрованного слова.

Во второй строке следует строка  $s$  длины  $n$ , состоящая из строчных букв латинского алфавита — зашифрованное слово.

#### Формат выходных данных

Выведите слово, которое зашифровал Поликарп.

#### Примеры

стандартный ввод	стандартный вывод
5 logva	volga
2 no	no
4 abba	baba

#### Замечание

В первом примере Поликарп зашифровал слово **volga**. Сначала он выписал букву **l**, стоящую в позиции 3, после чего его слово приняло вид **voga**. После этого, Поликарп выписал букву **o**, стоящую в позиции 2, а его слово стало равным **vga**. Затем, Поликарп выписал букву **g**, стоящую во второй позиции, после этого слово изменилось на **va**. Потом он выписал букву **v**, а затем букву **a**. Таким образом, зашифрованное слово выглядит как **logva**.

Во втором примере Поликарп зашифровал слово **no**. Сначала он выписал букву **n**, слово стало равным **o**, и он выписал букву **o**. Таким образом, в этом примере, слово и его зашифровка выглядят одинаково.

В третьем примере Поликарп зашифровал слово **baba**. Сначала он выписал букву **a**, стоящую в позиции 2, после чего слово стало равным **bba**. Затем он выписал букву **b**, стоящую в позиции 2, а его слово приняло вид **ba**. После этого он выписал сначала букву **b**, стоящую в позиции 1, слово приняло вид **a**, и он выписал эту букву **a**. Таким образом, зашифрованное слово выглядит как **abba**.