

А. Повторяем таблицу умножения

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

Алиса загадала натуральное число n и попросила Боба выписать столбик из таблицы умножения для числа n — то есть все произведения $n \times k$ для каждого k от 1 до 10.

Помогите Бобу — выведите столбик для числа n .

Формат входных данных

Единственная строка содержит целое число n ($1 \leq n \leq 100$) — число, которое загадала Алиса.

Формат выходных данных

Выведите столбик из таблицы умножения для числа n :

- необходимо вывести 10 строк;
- i -я строка должна выглядеть как « n x i = m », где m — это результат умножения n на i (обратите внимание, что здесь символ «x» — это латинская строчная буква «икс»);
- запрещено выводить лишние пробелы.

Ознакомьтесь с примерами ниже, чтобы лучше понять формат вывода.

Система оценки

Подгруппа	Дополнительные ограничения	Баллы	Необходимые подгруппы
1	$n \leq 5$	30	—
2	—	70	1

Пример

стандартный ввод	стандартный вывод
7	7 x 1 = 7 7 x 2 = 14 7 x 3 = 21 7 x 4 = 28 7 x 5 = 35 7 x 6 = 42 7 x 7 = 49 7 x 8 = 56 7 x 9 = 63 7 x 10 = 70

В. Сетевой протокол

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

Боб разрабатывает простой сетевой протокол для аутентификации сообщений. Он решил использовать хеш-функцию, которая для натурального числа n находит значение $(5^n \bmod 7)$.

5^n — это результат перемножения числа пять n раз. Например, $5^2 = 25$, $5^3 = 125$.

\bmod — это операция взятия остатка от деления. Например, $(7 \bmod 3) = 1$, $(11 \bmod 4) = 3$, $(14 \bmod 7) = 0$.

Таким образом, например, для $n = 2$ значение $(5^n \bmod 7)$ будет равно $(25 \bmod 7) = 4$.

Помогите Бобу — реализуйте данную хеш-функцию. На вход Вы будете получать число n , а на выход Вы должны выводить $(5^n \bmod 7)$.

Во многих языках программирования (например в Python и C++) операция взятия остатка от деления обозначается в виде символа `%`.

Формат входных данных

Единственная строка содержит целое число n ($1 \leq n \leq 10^{18}$).

Формат выходных данных

Выведите значение $(5^n \bmod 7)$.

Система оценки

Подгруппа	Дополнительные ограничения	Баллы	Необходимые подгруппы
1	$n \leq 10$	10	—
2	$n \leq 10^7$	40	1
3	—	50	1, 2

Примеры

стандартный ввод	стандартный вывод
2	4
100	2

С. Прямоугольники

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

У Алисы есть прямоугольник со сторонами A на B , а у Боба — прямоугольник со сторонами C на D .

Алиса и Боб очень любят квадраты, поэтому они хотят соединить свои прямоугольники таким образом, чтобы при объединении получился квадрат.

Правила, по которым соединяются прямоугольники:

- перед соединением прямоугольников их можно как угодно вращать;
- прямоугольники не могут накладываться друг на друга в каких-либо местах;
- прямоугольники могут соприкасаться.

Конечно, не всегда таким образом можно получить квадрат. В таком случае Алиса и Боб попробуют соединить свои прямоугольники так, чтобы получился новый прямоугольник. Если и это не удастся, Алиса и Боб сильно расстроятся и оставят попытки как-нибудь соединить свои прямоугольники.

Вычислите, в какую фигуру Алиса и Боб могут соединить свои прямоугольники.

Формат входных данных

Первая строка содержит целое число A ($1 \leq A \leq 100$).
Вторая строка содержит целое число B ($1 \leq B \leq 100$).
Третья строка содержит целое число C ($1 \leq C \leq 100$).
Четвёртая строка содержит целое число D ($1 \leq D \leq 100$).

Формат выходных данных

Выведите

- строку «SQUARE», если Алиса и Боб могут получить квадрат;
- строку «RECTANGLE», если Алиса и Боб не могут получить квадрат, но могут получить прямоугольник;
- строку «:(», если Алиса и Боб не могут получить ни квадрат, ни прямоугольник.

Система оценки

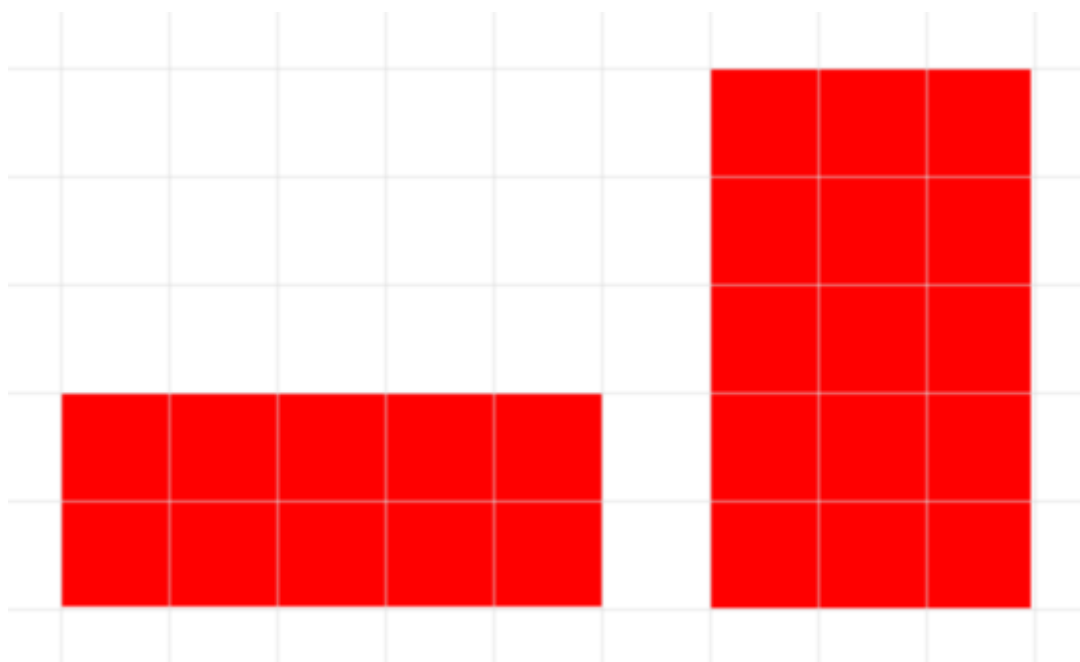
Подгруппа	Дополнительные ограничения	Баллы	Необходимые подгруппы
1	$A = B, C = D$	15	—
2	$A = B$	15	1
3	$A = C$	15	—
4	—	55	1, 2, 3

Примеры

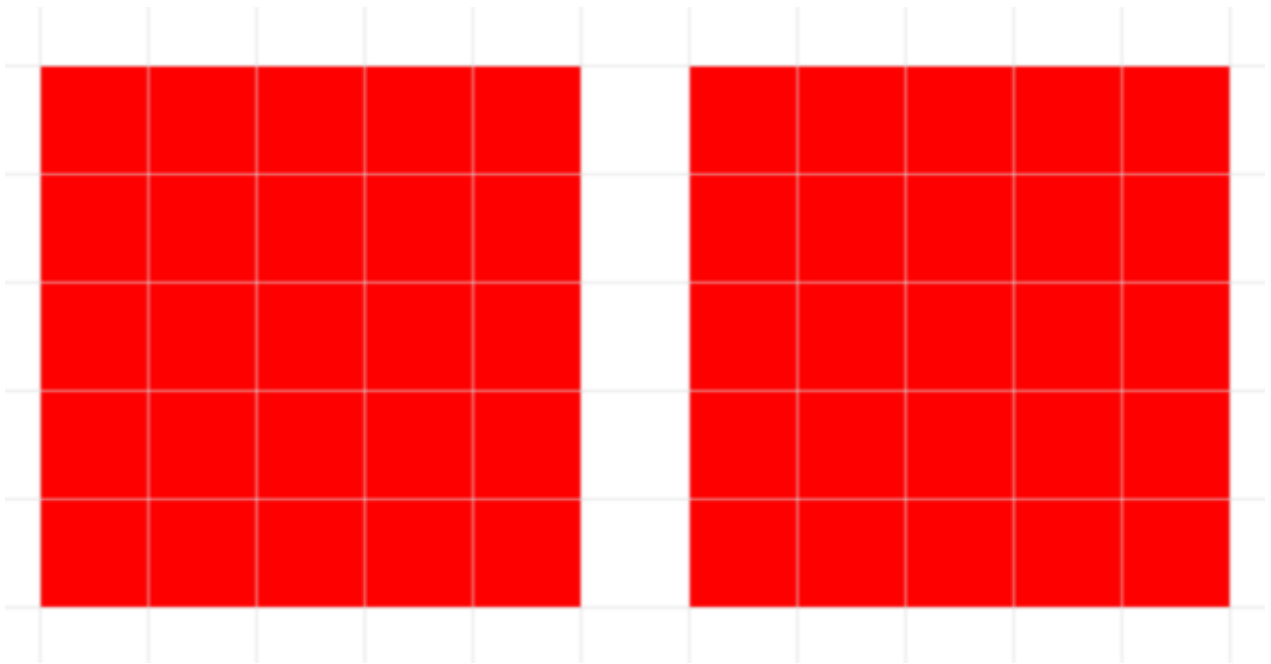
стандартный ввод	стандартный вывод
5 2 3 5	SQUARE
5 5 5 5	RECTANGLE
1 2 3 4	: (

Замечание

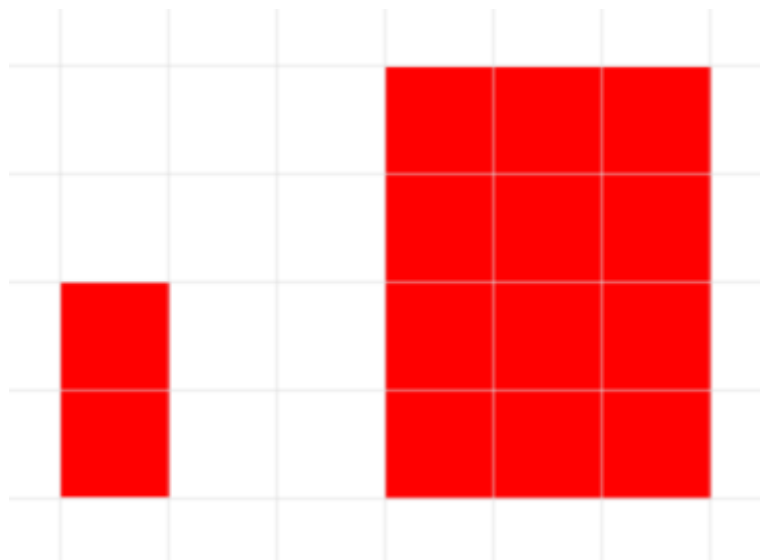
В первом примере можно повернуть прямоугольник Боба на 90 градусов, прямоугольник примет горизонтальное положение. Далее его можно пододвинуть сверху к прямоугольнику Алисы. Получится квадрат.



Во втором примере квадрат никак не получить, но сдвинув прямоугольники к друг другу можно получить новый прямоугольник.



В третьем примере нельзя получить ни квадрат, ни прямоугольник.



Д. Настольная игра

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

Алиса и Боб играют в настольную игру против друг друга. Боб уже на волоске от проигрыша. Сейчас ему предстоит бросить игральные кубики. Если на кубиках выпадут значения, сумма которых меньше числа X , то он окончательно проиграет.

Могли бы Вы быстренько посчитать для Боба, сколько существует различных исходов броска игровых кубиков, при которых он не проигрывает окончательно и у него остаётся шанс на победу?

Боб будет кидать n игровых кубиков. На каждом кубике независимо выпадает равновероятное значение от 1 до 6. Если Боб кидает несколько кубиков ($n > 1$), то он их кидает по очереди, а не все сразу.

Два исхода броска считаются различными, если значение, которое выпало на i -м кубике в первом исходе, отличается от значения, которое выпало на i -м кубике во втором исходе.

Например, пусть Боб кидает три кубика, тогда исходы «на первом кубике выпало 2, на втором выпало 6, на третьем выпало 5» и «на первом кубике выпало 2, на втором выпало 4, на третьем выпало 5» считаются различными, так как в первом исходе на втором кубике выпало 6, а во втором исходе на втором кубике выпало 4.

Исходы «на первом кубике выпало 2, на втором выпало 6, на третьем выпало 5» и «на первом кубике выпало 6, на втором выпало 2, на третьем выпало 5» — тоже различные, так как значения различаются на первом и втором кубиках.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 12$) — количество кубиков, которые будет кидать Боб.

Вторая строка содержит целое число X ($1 \leq X \leq 6 \times n$).

Формат выходных данных

Выведите количество различных исходов броска игровых кубиков, при которых у Боба остаётся шанс на победу.

Система оценки

Подгруппа	Дополнительные ограничения	Баллы	Необходимые подгруппы
1	$n = 1$	5	—
2	$n \leq 3$	10	1
3	$n \leq 8$	15	1, 2
4	$X \geq 6 \times n - 2$	20	—
5	—	50	1, 2, 3, 4

Примеры

стандартный ввод	стандартный вывод
2 10	6
3 2	216

Замечание

В первом примере Боб бросает 2 кубика. У него останутся шансы на победу, если у него выпадет сумма значений не менее 10. Всего 6 таких исходов:

- на первом кубике выпало 4, на втором 6;
- на первом кубике выпало 5, на втором 5;
- на первом кубике выпало 5, на втором 6;
- на первом кубике выпало 6, на втором 4;
- на первом кубике выпало 6, на втором 5;
- на первом кубике выпало 6, на втором 6;

Во втором примере Боб бросает 3 кубика. У него останутся шансы на победу, если у него выпадет сумма значений не менее 2. Несложно заметить, что при любом исходе броска кубиков сумма значений будет не меньше 3, значит подходят все возможные исходы. На каждом кубике возможны 6 вариантов значений, следовательно всего возможных исходов — $6 \times 6 \times 6 = 216$.

Е. PPSHneyneF4

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

Алиса и Боб придумали свой язык программирования PPSHneyneF4.

Данный язык очень прост, программа на PPSHneyneF4 может содержать команды всего двух видов:

- `print c` — вывести символ c , где c — строчная латинская буква;
- `call k` — вызвать команду с номером k .

Программа на PPSHneyneF4 выглядит как упорядоченный список из n команд. Команды пронумерованы числами от 1 до n и расположены в списке по возрастанию номера. При выполнении программы команды исполняются поочерёдно друг за другом по возрастанию своего номера. Команда с номером i ($i > 1$) не начнёт исполняться, пока не будет завершён процесс, который был вызван командой с номером $(i - 1)$.

Для примера рассмотрим работу следующей программы:

```
print a
call 4
call 2
print b
```

1. Исполняется **первая** команда «`print a`» — выводится символ «a». Текущий вывод: «a».
2. Исполняется **вторая** команда «`call 4`» — вызывается четвёртая команда «`print b`». Исполняется команда «`print b`» — выводится символ «b». Текущий вывод: «ab».
3. Исполняется **третья** команда «`call 2`» — вызывается вторая команда «`call 4`». Исполняется команда «`call 4`» — вызывается четвёртая команда «`print b`». Исполняется команда «`print b`» — выводится символ «b». Текущий вывод: «abb».
4. Исполняется последняя **четвёртая** команда — выводится символ «b». Текущий вывод: «abbb».
5. Окончательный вывод программы — «abbb».

Придумать язык программирования — это полдела, нужно ещё написать программу-интерпретатор, которая считывает программу на языке PPSHneyneF4, исполняет её команды и выводит результат работы.

Алиса и Боб нуждаются в Вашей помощи — реализуйте программу-интерпретатор.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 10^3$) — количество команд в программе на языке PPSHneyneF4.

Следующие n строк содержат список из n команд программы. Каждая команда приведена в отдельной строке.

Гарантируется, что существует хотя бы одна команда «`print`».

Формат выходных данных

- Если в процессе работы программы исполнится команда «call k », где $k < 1$ или $k > n$, то выполнение программы завершится с ошибкой и следует вывести «ERROR»;
- иначе, если процесс работы программы — бесконечный, то следует вывести «INFINITY»;
- в ином случае выведите в первой строке «OK», а во второй строке — результат работы программы.

Система оценки

Подгруппа	Дополнительные ограничения	Баллы	Необходимые подгруппы
1	$n = 1$	5	—
2	$n = 2$	10	1
3	Гарантируется, что программа состоит только из команд «print»	5	—
4	Гарантируется для каждой команды «call k », что команда с номером k — это «print»	15	3
5	Гарантируется, что процесс выполнения программы НЕ бесконечный	30	3, 4
6	—	35	1, 2, 3, 4, 5

Примеры

стандартный ввод	стандартный вывод
4 print a call 4 call 2 print b	OK abbb
2 print z call 10	ERROR
6 print h print e print l print l print o call 6	INFINITY

Замечание

Первый пример разобран в условии.

Во втором примере программа завершается с ошибкой, так как команда «call 10» вызывает команду с некорректным номером.

В третьем примере команда «call 6» бесконечно вызывает саму себя.

Е. Лабиринт

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

Алиса решила сыграть в компьютерную игру, где нужно играть за персонажа, который перемещается в лабиринте. Цель игры — добраться до горшочка с золотом, который где-то спрятан в лабиринте.

Эта задача оказалась непростой, Алисе даже кажется, что до горшочка невозможно добраться. Помогите Алисе — выясните, возможно ли из текущей позиции персонажа добраться до горшочка с золотом.

Формат входных данных

Если посмотреть на лабиринт сверху, то его можно представить как двумерное поле из клеток, которое имеет размеры в H рядов и W столбцов.

Первая строка входных данных содержит два целых числа H и W ($5 \leq H, W \leq 100$).

Следующие H строк описывают лабиринт. Каждая строка содержит W символов. Каждый символ описывает отдельную клетку:

- если символ равен «.», то клетка свободна;
- если символ равен «#», то клетка является стеной;
- если символ равен «A», то клетка свободна и в этой клетке находится текущая позиция персонажа;
- если символ равен «G», то клетка свободна и в этой клетке находится горшочек с золотом;
- других символов не существует в описании лабиринта.

Гарантируется, что в описании лабиринта

- крайние ряды и столбцы состоят только из символов «#»;
- существует ровно один символ «A»;
- существует ровно один символ «G».

Персонаж может переместиться из клетки A в клетку B , если

- клетка B — свободная;
- клетки A, B имеют общую сторону.

Гарантируется, что если можно добраться из клетки A в клетку B с помощью некоторого количества перемещений, где по пути ни одна клетка не повторяется, то существует ровно одна последовательность таких перемещений.

Формат выходных данных

Выведите «YES», если существует такая последовательность перемещений персонажа, после которых персонаж окажется в одной клетке с горшочком золота. Иначе выведите «NO».

Система оценки

Подгруппа	Дополнительные ограничения	Баллы	Необходимые подгруппы
1	Если возможно добраться до горшочка с золотом, то нужно сделать не более 2 перемещений	10	—
2	Если возможно добраться до горшочка с золотом, то нужно сделать не более 6 перемещений	20	1
3	—	70	1, 2

Примеры

стандартный ввод	стандартный вывод
<pre> 5 5 ##### #...# #.#.# #A#G# ##### </pre>	YES
<pre> 5 5 ##### #.G.# ##### #A..# ##### </pre>	NO
<pre> 9 13 ##### #.....# #...##### #..G#.#.#...# #...#.#...# #.#A.....# #.#.######.# #.#.#...#...# ##### </pre>	YES