

Problem A. Кузнечик

Input file: grig.in
Output file: grig.out
Time limit: 0.5 секунды
Memory limit: 256 Мебибайт

У одного из преподавателей кружка в комнате живёт кузнечик, который очень любит прыгать по клетчатой одномерной доске. Длина доски — N клеток. К его сожалению он умеет прыгать только на $1, 2, \dots, k$ клеток вперёд.

Однажды преподавателям стало интересно, сколькими способами кузнечик может допрыгать из первой клетки до последней. Помогите им ответить на этот вопрос.

Input

В первой и единственной строке входного файла записано два целых числа — N и k ($1 \leq N \leq 30, 1 \leq k \leq 10$).

Output

Выведите одно число — количество способов, которыми кузнечик может допрыгать из первой клетки до последней.

Example

grig.in	grig.out
8 2	21

Problem B. Рюкзак

Input file: knapsack.in
Output file: knapsack.out
Time limit: 0.5 секунды
Memory limit: 256 Мебибайт

Найдите максимальный вес золота, который можно унести в рюкзаке вместительностью S , если есть N золотых слитков с заданными весами.

Input

В первой строке входного файла записаны два числа — S и N ($1 \leq S \leq 10\,000, 1 \leq N \leq 300$). Далее следует N неотрицательных целых чисел, не превосходящих $100\,000$ — веса слитков.

Output

Выведите искомый максимальный вес.

Example

knapsack.in	knapsack.out
10 3 5 7 4	9

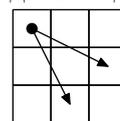
Problem C. Ход конём

Input file: knight.in
Output file: knight.out
Time limit: 0.5 секунды
Memory limit: 256 Мебибайт

Дана прямоугольная доска $N \times M$ (N строк и M столбцов).

В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

Input

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N, M \leq 50$).

Output

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

Example

knight.in	knight.out
3 2	1
31 34	293930

Problem D. Мутанты

Input file: mutants.in
Output file: mutants.out
Time limit: 0.5 секунды
Memory limit: 256 Мебибайт

Как вы помните, в Институте Искусств, Мутантов и Информационных Технологий разводят милых разноцветных зверюшек. Но вдруг одна из зверюшек нашла выход из Института и сбежала. По воле судьбы она попала в удивительный город Мутантоград. Вы не поверите, город разбит на улицы, на пересечении улиц находятся перекрестки.

Удивителен же Мутантоград тем, что ходить можно с перекрестка на перекресток только на восток или на юг, а также на каждом перекрестке берут штрафы. Наш мутант нашел карту города, она представляет собой клетчатый прямоугольник N на M , в котором на пересечении i -ой строки и j -ого столбца указан размер штрафа при попадании на этот перекресток.

Зверюшка находится на северо-западном углу города. Помогите ей дойти до юго-восточного угла Мутантограда, заплатив минимально возможный штраф.

Input

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N, M \leq 1000$).

В последующих N строках содержатся по M чисел — карта города Мутантоград.

Output

В первой строчке выведите одно целое число —

минимальный размер штрафа, который придется заплатить мутантику.

Во второй строчке выведите количество перекрестков на пути.

В следующих строчках выведите координаты перекрестков, через которые зверюшка пройдет. Гарантируется, что штраф не превысит 10^9 .

Example

mutants.in	mutants.out
3 4	35
5 9 4 3	6
3 1 6 9	1 1
8 6 8 12	2 1
	2 2
	3 2
	3 3
	3 4

Problem E. Плавные числа

Input file: numbers.in
Output file: numbers.out
Time limit: 0.5 секунды
Memory limit: 256 Мебибайт

Назовём натуральное число плавным, если разность любых двух его соседних цифр не превосходит по модулю единицы. Вам необходимо определить количество N -значных плавных чисел.

Input

В единственной строке входного файла одно число N ($1 \leq N \leq 20$).

Output

Вывести одно число — искомое количество плавных чисел.

Example

numbers.in	numbers.out
2	26

Problem F. Новая модель телефона

Input file: newphone.in
Output file: newphone.out
Time limit: 0.5 секунды
Memory limit: 256 Мебибайт

Компания Gnusmas разработала новую модель мобильного телефона. Основное достоинство этой модели — ударопрочность: её корпус сделан из особого сплава, и телефон должен выдерживать падение с большой высоты.

Компания Gnusmas арендовала n -этажное здание и наняла экспертов, чтобы те при помощи серии экспериментов выяснили, с какой высоты бросать телефон можно, а с какой — нельзя. Один эксперимент заключается в том, чтобы бросить телефон с какого-то этажа и посмотреть, сломается он от этого или нет. Известно, что любой

телефон этой модели ломается, если его сбросить с x -го этажа или выше, где x — некоторое целое число от 1 до n , включительно, и не ломается, если сбросить его с более низкого этажа. Задача экспертов заключается в том, чтобы узнать число x и передать его рекламному отделу компании.

Задача осложняется тем, что экспертам предоставлено всего k образцов новой модели телефона. Каждый телефон можно бросать сколько угодно раз, пока он не сломается; после этого использовать его для экспериментов больше не удастся.

Подумав, эксперты решили действовать так, чтобы минимизировать максимально возможное количество экспериментов, которое может потребоваться произвести. Чему равно это количество?

Input

В первой строке входного файла записаны через пробел два целых числа n и k — количество этажей в здании и количество предоставленных телефонов ($1 \leq n \leq 100\,000$, $0 \leq k < n$).

Output

В выходной файл выведите единственное число — минимальное количество экспериментов, которое потребуются совершить, чтобы узнать число x и использовать не более k телефонов. Если решить задачу невозможно, выведите вместо этого -1 .

Examples

newphone.in	newphone.out
4 2	2
4 1	3

В первом примере сначала следует бросить телефон со второго этажа. Если он сломается, то второй бросок следует сделать с первого этажа. В случае поломки станет известно, что $x = 1$. Иначе мы узнаем, что $x = 2$.

Если же при броске со второго этажа телефон не сломался, бросим телефон с третьего этажа. При поломке будет ясно, что $x = 3$. Иначе из условия $1 \leq x \leq 4$ следует, что $x = 4$.

Всего будет сделано два эксперимента. В них будет использовано не более чем два телефона.

Во втором примере следует сначала бросить единственный данный нам телефон с первого этажа, если он не сломается, то со второго, а если опять не сломается, то с третьего. При первой же поломке мы узнаем точное значение x . Если после трёх бросков телефон так и не сломался, то $x = 4$.