

Codeforces Round #289 (Div. 2, ACM ICPC Rules)

А. Максимум в таблице

ограничение по времени на тест: 2 секунды
 ограничение по памяти на тест: 256 мегабайт
 ввод: стандартный ввод
 вывод: стандартный вывод

Таблица a размера $n \times n$ задаётся следующим образом:

- В первой строке и в первом столбце таблицы стоят единицы, то есть $a_{i,1} = a_{1,i} = 1$ для всех $i = 1, 2, \dots, n$.
- Каждое из оставшихся чисел в таблице определяется как сумма числа над ним и числа слева от него. Иными словами, остальные элементы таблицы задаются формулой $a_{i,j} = a_{i-1,j} + a_{i,j-1}$.

Эти условия однозначно задают все элементы таблицы.

Вам даётся число n . Требуется определить значение максимального числа в таблице $n \times n$, построенной по вышеприведенным правилам.

Входные данные

В единственной строке входных данных задано натуральное число n ($1 \leq n \leq 10$) — количество строк и столбцов таблицы.

Выходные данные

В единственной строке выведите натуральное число m — максимальное число построенной таблицы.

Примеры тестов

входные данные
1
выходные данные
1
входные данные
5
выходные данные
70

Примечание

Во втором тесте строки таблицы выглядят так:

$\{1, 1, 1, 1, 1\},$
 $\{1, 2, 3, 4, 5\},$
 $\{1, 3, 6, 10, 15\},$
 $\{1, 4, 10, 20, 35\},$
 $\{1, 5, 15, 35, 70\}.$

В. Раскраска шаров

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

На столе лежит n кучек камней, в i -й кучке находится a_i камней. Требуется раскрасить каждый камень в один из k цветов так, чтобы для каждого цвета c и для любых двух кучек i и j количества камней цвета c в кучках i и j отличались не более чем на один.

Иными словами, пусть $b_{i,c}$ — количество камней в i -й кучке, покрашенных в цвет c . Тогда для любых $1 \leq c \leq k$, $1 \leq i, j \leq n$ должно выполняться $|b_{i,c} - b_{j,c}| \leq 1$. Необязательно использовать все k цветов: если цвет c не встречается в кучке i , то $b_{i,c}$ полагается равным нулю.

Входные данные

В первой строке входных данных заданы натуральные числа n и k ($1 \leq n, k \leq 100$), разделенные пробелом — количество кучек и количество цветов соответственно.

Во второй строке заданы n натуральных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$), задающие количества камней в кучках.

Выходные данные

Если раскраски камней, удовлетворяющей условию задачи, не существует, в единственной строке выведите «NO» (без кавычек).

Иначе в первой строке «YES» (без кавычек). Далее должны следовать n строк, в i -й из них должны находиться a_i чисел, разделенных пробелами. j -е ($1 \leq j \leq a_i$) из этих чисел должно равняться цвету j -го камня в i -й кучке. **Если возможных ответов несколько, разрешается вывести любой.**

Примеры тестов

входные данные
4 4 1 2 3 4
выходные данные
YES 1 1 4 1 2 4 1 2 3 4

входные данные
5 2 3 2 4 1 3
выходные данные
NO

входные данные
5 4 3 2 4 3 5
выходные данные
YES 1 2 3 1 3 1 2 3 4 1 3 4 1 1 2 3 4

С. Суммы цифр

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

У Васи была **строго возрастающая** последовательность целых положительных чисел a_1, \dots, a_n . Вася построил по ней новую последовательность b_1, \dots, b_n , где b_i — сумма цифр a_i в десятичной записи. После этого последовательность a_i потерялась, осталась только последовательность b_i .

Васе интересно, какими могли быть числа a_i . Из всех вариантов последовательности a ему интересен такой, в котором последнее число a_n является минимально возможным. Помогите Васе восстановить исходную последовательность.

Гарантируется, что подобная последовательность всегда существует.

Входные данные

В первой строке записано одно целое число n ($1 \leq n \leq 300$).

В следующих n строках записаны числа b_1, \dots, b_n — требуемые суммы цифр. Все b_i удовлетворяют ограничениям $1 \leq b_i \leq 300$.

Выходные данные

Выведите n чисел по одному на строке — корректный вариант для чисел a_i , в порядке возрастания индексов. Последовательность должна быть строго возрастающей. Сумма цифр i -го числа должна быть равна b_i . Если вариантов с минимальным значением последнего числа несколько, выведите любой. Числа следует выводить без ведущих нулей.

Примеры тестов

входные данные	
3	
1	
2	
3	
выходные данные	
1	
2	
3	

входные данные	
3	
3	
2	
1	
выходные данные	
3	
11	
100	

D. Восстановление чисел

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

У Васи было два массива, составленных из целых неотрицательных чисел: a размера n и b размера m . Вася выбрал натуральное число k и построил таблицу v размера $n \times m$ по следующему правилу:

$$v_{i,j} = (a_i + b_j) \bmod k$$

Вася записал таблицу v на листок и положил в стол.

Спустя год Вася, копаясь в кипе вещей в столе, нашёл какой-то листок, на котором была записана таблица w размера $n \times m$. Он вспомнил, что когда-то давно строил таблицу по вышеописанным правилам, но он не уверен, что нашёл тот самый листок с таблицей v . Ваша задача — выяснить, могла ли найденная таблица w быть получена по этим правилам, и если да, то определить какой-нибудь набор чисел $k, a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m$, по которым могла получиться такая таблица.

Входные данные

В первой строке заданы натуральные числа n и m ($1 \leq n, m \leq 100$), разделенные пробелом — количество строк и столбцов в найденной таблице соответственно.

В i -й из последующих строк следуют числа $w_{i,1}, w_{i,2}, \dots, w_{i,m}$ ($0 \leq w_{i,j} \leq 10^9$), разделенные пробелами — элементы i -й строки таблицы w .

Выходные данные

Если таблица w не могла быть получена указанным в условии способом, в единственной строке выведите «NO» (без кавычек).

Иначе нужно вывести четыре строки.

В первой строке выведите «YES» (без кавычек).

Во второй строке выведите натуральное число k ($1 \leq k \leq 10^{18}$). Обратите внимание, что каждый элемент таблицы w должен лежать в пределах от 0 до $k - 1$ включительно.

В третьей строке выведите n натуральных чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^{18}$).

В четвертой строке выведите m натуральных чисел b_1, b_2, \dots, b_m ($0 \leq b_i \leq 10^{18}$).

Примеры тестов

входные данные
2 3 1 2 3 2 3 4
выходные данные
YES 1000000007 0 1 1 2 3

входные данные
2 2 1 2 2 0
выходные данные
YES 3 0 1 1 2

входные данные
2 2 1 2 2 1
выходные данные
NO

Примечание

Под записью $b \bmod c$ подразумевается остаток от целочисленного деления b на c .

Гарантируется, что если существует какой-то набор чисел $k, a_1, \dots, a_n, b_1, \dots, b_m$, по которому могла быть построена таблица w , то существует и набор чисел, удовлетворяющий ограничениям $1 \leq k \leq 10^{18}$, $1 \leq a_i \leq 10^{18}$, $1 \leq b_i \leq 10^{18}$. Иными словами, эти верхние границы введены только для удобства проверки.

Е. Мелодичная песня

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

В седьмом классе Саша начал слушать музыку. Для того чтобы оценивать, какая из песен нравится ему больше, он ввел понятие *мелодичности* песни. Название песни — это слово из заглавных латинских букв. *Мелодичность* песни это *мелодичность* её названия.

Назовем *простой мелодичностью* слова отношение количества гласных букв в слове к количеству всех букв в слове.

Назовем *мелодичностью* слова сумму *простых мелодичностей* всех подстрок слова.

Более формально, определим функцию $vowel(c)$, равную 1, если c — гласная, и 0 иначе. Пусть s_i — i -й символ строки s , а $s_{i..j}$ — подстрока слова s , начинающаяся с i -го символа и заканчивающаяся j -м символом ($s_{isi+1} \dots s_j, i \leq j$).

Тогда *простая мелодичность* s определяется по формуле:

$$simple(s) = \frac{\sum_{i=1}^{|s|} vowel(s_i)}{|s|}$$

Мелодичность s равна

$$\sum_{1 \leq i \leq j \leq |s|} simple(s_{i..j}).$$

Найдите мелодичность данной песни.

Гласными буквами считаются I, E, A, O, U, Y .

Входные данные

На ввод подаётся единственная строка s ($1 \leq |s| \leq 5 \cdot 10^5$) — название песни.

Выходные данные

Определите *мелодичность* песни с абсолютной или относительной погрешностью не более 10^{-6} .

Примеры тестов

входные данные
IEAIAIO
выходные данные
28.0000000

входные данные
BYOB
выходные данные
5.8333333

входные данные
YISVOWEL
выходные данные
17.0500000

Примечание

В первом примере все буквы — гласные. *Простая мелодичность* каждой подстроки равна 1.

Всего в слове длины 7 имеется 28 подстрок. Значит, *мелодичность* песни равна 28.

Ф. Контроль успеваемости

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

На одной из своих контрольных работ по информатике Дмитрий Олегович решил дать следующую задачу:

Пусть дано дерево T на n вершинах, заданное своей матрицей смежности $a[1 \dots n, 1 \dots n]$. Какую последовательность чисел выведет следующий псевдокод?

```
used[1 ... n] = {0, ..., 0};

procedure dfs(v):
    print v;
    used[v] = 1;
    for i = 1, 2, ..., n:
        if (a[v][i] == 1 and used[i] == 0):
            dfs(i);

dfs(1);
```

Чтобы было легче проверять работы студентов, Дмитрий Олегович решил придумать такое дерево T , чтобы в ответе получилась его любимая последовательность b . С другой стороны, Дмитрий Олегович не хочет давать студентам одинаковые данные в контрольной работе — ведь в таком случае не миновать списывания. Поэтому у Дмитрия Олеговича возник следующий вопрос: сколько существует таких деревьев, что при запуске данного псевдокода на них выведенная последовательность в точности совпадает с последовательностью b ?

Два дерева на n вершинах считаются различными, если их матрицы смежности a_1 и a_2 не совпадают, то есть существует такая пара (i, j) , что $1 \leq i, j \leq n$ и $a_1[i][j] \neq a_2[i][j]$.

Входные данные

В первой строке задано натуральное число n ($1 \leq n \leq 500$) — количество чисел в последовательности b .

Во второй строке задаются n натуральных чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq n$), разделенные пробелом. Гарантируется, что последовательность b является перестановкой. Иными словами, каждое из чисел $1, 2, \dots, n$ встречается в последовательности b ровно один раз. Также гарантируется, что $b_1 = 1$.

Выходные данные

Выведите одно число — остаток от деления количества подходящих деревьев на $10^9 + 7$.

Примеры тестов

входные данные
3 1 2 3
выходные данные
2
входные данные
3 1 3 2
выходные данные
1