

Функция Гранди

Код задачи: FGRANDY

Время на один тест: 1 сек

Пусть $G=(V, E)$ – произвольный ориентированный граф. Для каждой вершины x этого графа введем множество $Next(x)$ как множество всех вершин, в которые ведут дуги, выходящие из вершины x .

Пусть F – функция, принимающая целые неотрицательные значения, определенная на вершинах графа G . Функция F называется функцией Гранди, если для каждой вершины x графа G выполняется условие $F(x)=\min\{n \geq 0 \mid n \neq F(y) \text{ ни для какой вершины } y \text{ из } Next(x)\}$.

Рассмотрим класс ориентированных графов, у которых в каждую вершину входит ровно одна дуга. Напишите программу, которая получает на вход некоторый граф из описанного выше класса и находит функцию Гранди для этого графа, если она существует.

Входной файл FGRANDY.IN содержит описание ориентированного графа. В первой строке файла находится целое число N ($2 \leq N \leq 100000$), равное количеству вершин графа. В i -й из следующих N строк находится одно целое число P_i ($1 \leq P_i \leq N$; $P_i \neq i$), равное номеру вершины, из которой выходит единственная дуга, ведущая в вершину с номером i .

Если для графа, описанного во входном файле, не существует функции Гранди, то выходной файл FGRANDY.OUT должен содержать одну строку со словом 'NO'. В противном случае, файл должен состоять из N строк, в i -й из которых должно быть записано одно целое число F_i – значение найденной функции Гранди для вершины с номером i . Если существует несколько функций Гранди для графа, описанного во входном файле, то выведите любую из них.

Пример входных данных #14
2
4
4
1Пример выходных данных #10
1
0
2Пример входных данных #23
3
1
2Пример выходных данных #2

NO

Вычеркивание двух

Код задачи: CUTTWO

Время на один тест: 1 сек

Задана строка S , состоящая из маленьких букв латинского алфавита. Сколько различных строк можно получить при помощи вычеркивания ровно двух символов из S ?

Входной файл CUTTWO.IN содержит строку S , записанную в первой и единственной строке файла. Длина строки S от 2 до 100000 символов включительно. Строка S содержит только маленькие буквы латинского алфавита.

Выходной файл CUTTWO.OUT должен содержать одно целое число, равное количеству различных строк, которые можно получить при помощи вычеркивания ровно двух символов из S .

Пример входных данных #1

abbccc

Пример выходных данных #1

5

Красивый город

Код задачи: MAYOR

Время на один тест: 1 сек

Мэр одного небольшого города очень хочет, чтобы его город был как можно более красивым и готов на многое пойти для достижения этой цели.

Карту города можно представить себе как матрицу из N строк и M столбцов, в каждой клетке которой находится один дом. Каждый дом характеризуется целым числом, называемым его значением красоты. Значение красоты дома, который расположен на пересечении i -й строки и j -го столбца матрицы, равно A_{ij} .

Чтобы сделать город красивее, мэр решил разрушить некоторые дома. Мэр планирует провести некоторое количество операций разрушения, каждая из которых заключается в том, что все дома в некоторой строке или некотором столбце матрицы уничтожаются.

Известно, что значение красоты города может быть вычислено по формуле $B=S-K*C$, где B – значение красоты города, S – сумма значений красоты всех неразрушенных домов в городе, K – количество проведенных операций разрушения, C – некоторое целое число, называемое коэффициентом разрушения.

Помогите мэру достичь максимального возможного значения красоты города путем проведения некоторого неотрицательного количества операций разрушения.

Входной файл MAYOR.IN содержит описание города. В первой строке файла находятся целые числа N , M и C ($1 \leq N$, $M \leq 20$, $0 \leq C \leq 100000$), разделенные пробелами. Каждая из оставшихся N строк входного файла содержит M целых чисел, разделенных пробелами. j -е число в i -й из этих строк равно A_{ij} ($-10000 \leq A_{ij} \leq 10000$).

Выходной файл MAYOR.OUT должен содержать одно целое число, равное максимальному возможному значению красоты города, которого возможно достичь путем проведения некоторого неотрицательного количества операций разрушения.

Пример входных данных #1

```
3 5 4
-1 -1 -1 -1 -1
-1 1 -1 1 -1
-1 -1 -1 -1 -1
```

Пример выходных данных #1

```
-9
```

Диаграмма Вороного

Код задачи: VORONOI

Время на один тест: 1 сек

Рассмотрим декартову прямоугольную систему координат Oxy , на которой для измерения расстояния между точками используется манхэттенская метрика, т.е. расстояние между точками (x', y') и (x'', y'') равно $|x'-x''|+|y'-y''|$.

Пусть S – квадрат с вершинами в точках $(0, 0)$, $(0, A)$, $(A, 0)$, (A, A) , т.е. $S = \{(x, y) | 0 \leq x \leq A \text{ и } 0 \leq y \leq A\}$, а $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ – N различных точек, принадлежащих этому квадрату, т.е. $(x_i, y_i) \in S$ для любого i от 1 до N .

Обозначим через $d_i(x, y)$ расстояние между точками (x, y) и (x_i, y_i) , т.е. $d_i(x, y) = |x_i - x| + |y_i - y|$, а через $d_{\min}(x, y)$ минимальное из расстояний от точки (x, y) до точек (x_i, y_i) , т.е. $d_{\min}(x, y) = \min\{d_i(x, y) | i=1, 2, \dots, N\}$.

Областью Вороного для точки (x_i, y_i) называется множество $V_i = \{(x, y) \in S | d_i(x, y) = d_{\min}(x, y)\}$.

Напишите программу, которая для каждой точки (x_i, y_i) вычисляет площадь области Вороного V_i .

В первой строке входного файла VORONOI.IN записаны два целых числа A и N ($1 \leq A \leq 100$, $1 \leq N \leq 100$), разделенные пробелом. В i -й из следующих N строк файла записаны два целых числа x_i и y_i ($0 \leq x_i, y_i \leq A$), разделенные пробелом.

Выходной файл VORONOI.OUT должен состоять из N строк. В i -й строке выходного файла должно быть записано одно вещественное число, равное найденному Вашей программой значению площади области Вороного V_i . Абсолютная погрешность найденного Вашей программой значения V_i не должна превышать 10^{-5} .

Пример входных данных #1

```
4 4
1 1
3 1
3 3
1 3
```

Пример выходных данных #1

```
4.000000000000
4.000000000000
4.000000000000
4.000000000000
```

ENHADINUMS mod P

Код задачи: EDNP

Время на один тест: 1 сек

ENHADINUMS mod P (сокращенно, EDNP) – это модифицированная версия формата записи десятичных чисел ENHADINUMS (Enhanced Digital Numbers), который придумал Gerson Kurz из Германии.

Формат EDNP позволяет записывать десятичные числа от 0 до P-1, где P – некоторое фиксированное натуральное число. Запись десятичного числа в формате EDNP представляет собой строку, выводимую из нетерминала NUMBER в следующей грамматике:

NUMBER → NUMERIC | NUMERIC NUMBER
 NUMERIC → SHIFT | DIGIT | NUMPLC
 SHIFT → '(' NUMBER '=' NUMBER ')'
 DIGIT → '|' | '+' | '#' | '.' | ':'
 NUMPLC → '"' | NUMPLCIN | ','
 NUMPLCIN → DIGIT | SHIFT | NUMPLCIN DIGIT | NUMPLCIN SHIFT

Свяжем с каждым нетерминалом описанной грамматики атрибут dec, который представляет собой целое число от 0 до P-1. Атрибут dec вычисляется по следующим правилам:

Продукция	Семантическое правило
NUMBER → NUMERIC	NUMBER.dec = NUMERIC.dec
NUMBER → NUMERIC NUMBER ₁	NUMBER.dec = (NUMERIC.dec+NUMBER ₁ .dec) mod P
NUMERIC → SHIFT	NUMERIC.dec = SHIFT.dec
NUMERIC → DIGIT	NUMERIC.dec = DIGIT.dec
NUMERIC → NUMPLC	NUMERIC.dec = NUMPLC.dec
SHIFT → '(' NUMBER ₁ '=' NUMBER ')'	SHIFT.dec = (NUMBER ₁ .dec*2 ^{NUMBER.dec}) mod P
DIGIT → ' '	DIGIT.dec = 1
DIGIT → '+'	DIGIT.dec = 2
DIGIT → '#'	DIGIT.dec = 4
DIGIT → '.'	DIGIT.dec = P-1
DIGIT → ':'	DIGIT.dec = P-2
NUMPLC → '"' NUMPLCIN ','	NUMPLC.dec = NUMPLCIN.dec
NUMPLCIN → DIGIT	NUMPLCIN.dec = DIGIT.dec

NUMPLCIN → SHIFT	NUMPLCIN.dec = SHIFT.dec
NUMPLCIN → NUMPLCIN ₁ DIGIT	NUMPLCIN.dec = (5*NUMPLCIN ₁ .dec+DIGIT.dec) mod P
NUMPLCIN → NUMPLCIN ₁ SHIFT	NUMPLCIN.dec = (5*NUMPLCIN ₁ .dec+SHIFT.dec) mod P

Пусть S – некоторая строка в формате EDNP. Для нахождения десятичного числа, записью которого является строка S, необходимо провести синтаксический разбор строки S и вычислить значение атрибута dec для каждого нетерминала в дереве разбора. Значение атрибута dec для нетерминала NUMBER в корне дерева разбора и будет тем десятичным числом, записью которого является строка S.

Напишите программу, которая вводит из входного файла целое число P и строку S, являющуюся записью некоторого десятичного числа в формате EDNP, и находит десятичное число, записью которого является строка S.

Входной файл EDNP.IN состоит из двух строк. В первой строке файла записано целое число P (7 ≤ P ≤ 1000). Во второй строке файла записана строка S, содержащая от 1 до 1000 символов включительно. Строка S состоит только из символов '|', '+', '#', '.', ':', '"', ',', '(', '=', ')' и является записью некоторого десятичного числа в формате EDNP (т.е. может быть выведена из нетерминала NUMBER в описанной выше грамматике).

Выходной файл EDNP.OUT должен содержать целое число от 0 до P-1, записью которого является строка S.

Пример входных данных #1

421
' . (| = #) # . ,

Пример выходных данных #1

294

Остаточный код

Код задачи: MODCODE

Время на один тест: 1 сек

Пусть $N = (a_1a_2\dots a_p)_P$ – P -значное число без лидирующих нулей, записанное в системе счисления с основанием P . Обозначим остаток от деления числа $(a_1a_2\dots a_i)_P$ на i через b_i для всех i от 1 до P . Вектор (b_1, b_2, \dots, b_p) будем называть остаточным кодом числа N .

Например, $P = 4$ и $N = (3102)_4$. Тогда остаточный код числа N равен $(0, 1, 1, 2)$. Существует еще 7 чисел, имеющих точно такой же остаточный код – это $(1122)_4, (1302)_4, (1332)_4, (2112)_4, (2322)_4, (3132)_4, (3312)_4$.

Напишите программу, которая вычисляет количество P -значных чисел без лидирующих нулей, записанных в системе счисления с основанием P , имеющих заданный остаточный код (b_1, b_2, \dots, b_p) .

Входной файл MODCODE.IN описывает некоторый остаточный код. Первая строка этого файла содержит одно целое число P ($2 \leq P \leq 16$), равное количеству элементов описываемого кода. i -я из следующих P строк входного файла содержит одно целое число b_i ($0 \leq b_i < i$), равное i -му элементу описываемого остаточного кода.

Выходной файл MODCODE.OUT должен содержать одно целое число, равное количеству P -значных чисел без лидирующих нулей, записанных в системе счисления с основанием P , имеющих остаточный код (b_1, b_2, \dots, b_p) .

Пример входных данных #1

4
0
1
1
2

Пример выходных данных #1

8

Подарок

Код задачи: PRESENT

Время на один тест: 1 сек

Скоро у мальчика Пети будет день рождения. Петя коллекционирует почтовые марки, поэтому его друзья решили подарить ему на день рождения A марок. В местном почтовом отделении марки продаются только в упаковках. Каждая упаковка содержит ровно B марок и стоит C рублей. Какую минимальную сумму денег необходимо иметь друзьям Пети, чтобы сделать подарок из A марок?

При решении задачи примите к сведению, что:

- Друзьям Пети не обязательно покупать ровно A марок. Они могут приобрести большее количество марок, и часть из них оставить себе.
- Число A может быть равно нулю. Это означает, что друзья Пети решили не дарить Пете ни одной марки.
- Число B может быть равно нулю. Это означает, что в упаковках марок, которые продаются на почте, на самом деле нет ни одной марки.
- Число C может быть равно нулю. Это означает, что упаковки с марками выдают на почте бесплатно.

Входной файл PRESENT.IN содержит одну строку, в которой записаны три целых числа A, B, C ($0 \leq A, B, C \leq 10000$) в указанном порядке. Числа во входном файле разделены пробелами.

Выходной файл PRESENT.OUT должен содержать одно целое число, равное минимальной сумме денег в рублях, которую необходимо иметь друзьям Пети, чтобы сделать подарок из A марок. В случае, когда сделать подарок из A марок невозможно, выходной файл должен содержать одну строку со словом 'IMPOSSIBLE'.

Пример входных данных #1

10 4 7

Пример выходных данных #1

21

Пример входных данных #2

10 0 7

Пример выходных данных #2

IMPOSSIBLE