

1. Приближение вещественного числа

Ограничение времени: 1.0 секунда

Ограничение памяти: 256 МБ

Задано вещественное положительное число A ($A \in R_+$) и $100 < A < 1000$. Приближением числа A назовем число $\bar{A} = a + b\sqrt{p}$, где a, b - целые числа из интервала $[2; \left\lceil \frac{A}{2} \right\rceil]$, $\left\lceil \frac{A}{2} \right\rceil$ - целая часть числа $\frac{A}{2}$, p - число из множества $P = \{2, 3, 5, 7, 11, 17\}$ и такое, что квадрат разности $\delta = (A - \bar{A})^2$ минимальный для всех для всех значений a, b и p из заданных для них интервалов и множества.

Требуется составить программу `Approached_p`, которая для заданного числа A_+ находит такие значения целых чисел a, b , при которых число $a + b\sqrt{p}$ наилучшим приближением числа A для всех $p \in P$.

Исходные данные

Строка ввода содержит значение вещественного положительного числа A .

Результат

Строка вывода включает значения чисел a, b, p, δ .

Пример (тест)

исходные данные	результат	Примечание
13.237	2 5 5 0.000128	$13.237 \approx 2 + 5\sqrt{5}$
179.537	79 38 7 17e^-11	$179.537 \approx 79 + 38\sqrt{7}$
739.473	53 308 5 18e^-14	$739.473 \approx 53 + 308\sqrt{5}$

2. Перестановка

Ограничение времени: 1.0 секунда

Ограничение памяти: 256 МБ

Заданы последовательности последовательности из n целых чисел:

$$a_1, a_2, \dots, a_n \text{ и } b_1, b_2, \dots, b_n, \quad i = \overline{1, n}, \quad a_i, b_i > 0, \quad b_i \geq a_i.$$

Построим последовательность чисел $c_i = \frac{a_i}{b_i}$, элементы которой пронумерованы тем же индексом i ($i = \overline{1, n}$). Упорядочение чисел c_i по возрастанию создает новую последовательность чисел c'_j пронумерованных индексом j ($j = \overline{1, n}$).

Таким образом, упорядочение элементов последовательности задает отображение исходной нумерации по индексу i к новой последовательности номеров j_k ($k = \overline{1, n}$),

где исходные номера i элементов последовательности $c_i = \frac{a_i}{b_i}$ переставлены в другом порядке, т. е. образуют перестановку

$$\begin{pmatrix} 1, 2, 3, \dots, n-1, n \\ j_1, j_2, j_3, \dots, j_{n-1}, j_n \end{pmatrix}.$$

Требуется составить программу `Rearrangement`, которая по заданному n и последовательностям a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n , $i = \overline{1, n}$, $a_i, b_i > 0$, $b_i \geq a_i$

выводит две последовательности чисел: значения начальных индексов элементов $c_i = \frac{a_i}{b_i}$ и перестановку их начальных индексов после упорядочения элементов последовательности в

порядке возрастания.

Пример

i	a_i	b_i	c_i	c'_j	j_k
1	2	7	2/7	3/13	2
2	3	13	3/13	1/4	4
3	5	9	5/9	2/7	1
4	1	4	1/4	5/9	3

Перестановка

$$\begin{pmatrix} 1, 2, 3, 4 \\ 2, 4, 1, 3 \end{pmatrix}$$

Исходные данные

Первая строка ввода — число n , вторая строка ввода — значения чисел последовательности a_1, a_2, \dots, a_n , третья строка ввода — значения чисел последовательности и b_1, b_2, \dots, b_n , где $i = \overline{1, n}$, $a_i, b_i > 0$, $b_i \geq a_i$

Результат

Первая строка вывода — последовательности чисел от 1 до n .

Вторая строка вывода — последовательность из перестановки чисел от 1 до n .

Пример (тест)

исходные данные	результат
4	1 2 3 4
2 3 5 1	2 4 1 3
7 13 9 4	

3. Принцип «Домино»

Ограничение времени: 1.0 секунда

Ограничение памяти: 256 МБ

Заданы наборы A_1 и A_2 пар чисел $\alpha_{ab} = (a; b)$, где $a, b \in \{0, 1, 2, 3, 4, 5, 6\}$.

Пары $(a; b)$ и $(b; a)$ считаются одинаковыми и в каждом из наборов может быть только одна из пар $(a; b)$ или $(b; a)$. Количество пар в каждом наборе задается числами n и m соответственно.

В каждый из заданных наборов пар входят только различные из пары $(a; b)$;

Назовем **прямой цепочкой** пар из наборов A_1 и A_2 и обозначим её $(a, b) \rightarrow (b, c)$, где $(a; b) \in A_1$, $(b, c) \in A_2$.

Назовем **обратной цепочкой** пар из наборов A_2 и A_1 и обозначим её $(c, a) \leftarrow (a, b)$, где $(c, a) \in A_2$, $(a, b) \in A_1$.

Пример.

Первый набор пар: $A_1 = \{(3; 2), (5; 2), (6; 3), (5; 4)\}$, $n = 4$, второй набор пар: $A_2 = \{(4; 2), (2; 6), (5; 5), (6; 0)\}$, $m = 4$.

Прямые цепочки: $(3; 2) \rightarrow (2; 6)$, $(5; 2) \rightarrow (2; 6)$, $(5; 4) \rightarrow (4; 2)$ — 3.

Обратные цепочки: $(5; 5) \leftarrow (5; 2)$, $(2; 6) \leftarrow (6; 3)$, $(5; 5) \leftarrow (5; 4)$ — 3.

Требуется составить программу domino, которая для заданных n и m и наборов пар вычисляет количество прямых и обратных цепочек.

Исходные данные

Первая строка ввода — значение числа n , вторая строка ввода — значения пар чисел

ФГБОУ ВПО Московский педагогический государственный университет
 Математический факультет
Кафедра теоретической информатики и дискретной математики
Задачи студенческой олимпиады по программированию 2016 г.

первого набора A_1 , третья строка ввода — значение числа m , четвертая строка — значения пар чисел второго набора A_2 .

Результат

Строка вывода, содержащая два числовых значения, соответственно, количество прямых и обратных цепочек.

Пример (тест)

исходные данные	результат
4	3 3
3 2 5 2 6 3 5 4	
4	
4 2 2 6 5 5 6 0	

4. Декодирование сообщения

Ограничение времени: 1.0 секунда
 Ограничение памяти: 256 МБ

Дана следующая таблица кодировки

Символ	Код	Знак	Код
A	- - -	Точка
B	- . . .	@	. - - - . - .
C	-	Тире —	- -
D	- . .		

Получатель получает закодированное кодом Морзе сообщение об адресе электронной почты отправителя. При этом запись закодированного адреса содержит только коды символов из таблицы. Структура адреса электронной почты стандартная :

<имя получателя>@<домен> ,

где <имя получателя> и <домен> могут содержать символы «точка» и «тире».

Требуется составить программу Decoding, которая декодирует получаемую символьную строку кодов Морзе в набор символов и знаков.

Пример

Исходное сообщение:

- - . . - . . . - - . - . - - . - . - - . - - . . - . - .

Декодированное сообщение: B . D - A @ A A . D C

Исходные данные

Строка ввода содержит тест закодированного сообщения в виде последовательности из символов точка и тире.

Результат

Строка вывода содержит текст исходного сообщения в виде последовательности символов и знаков, декодированных в соответствии с таблицей их кодов.

5. Угол и точка

Ограничение времени: 1.0 секунда

Ограничение памяти: 256 МБ

В первом квадранте декартовой системы координат задан квадрат с длиной стороны равной a и вершиной в начале координат. Заданы координаты внутренней точки L квадрата. Из точки L исходят два луча, направление которых определяют два заданных своими координатами вектора \vec{u}_1 и \vec{u}_2 ($\vec{u}_1 \neq \vec{u}_2$), исходящие из начала координат.

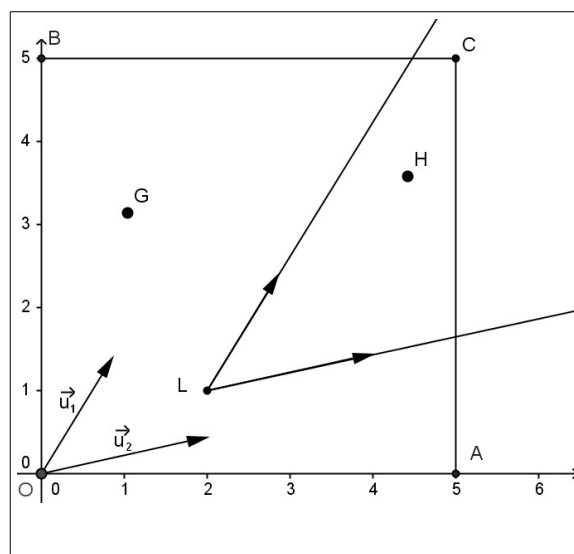
Требуется составить программу `Point_in_Corner`, которая для точки X , заданной своими координатами, определит лежит ли эта точка во внутренней или во внешней области угла.

Пример

Составим рисунок для следующих исходных данных: $a=5$, $L(2;1)$, $\vec{u}_1(0,88;1,42)$, $\vec{u}_2(2,04;0,44)$.

Точка $X=H(4,42;3,58)$ лежит во внутренней (inside) области угла, образованного лучами, исходящими из точки L и направление которых определяют два вектора \vec{u}_1 и \vec{u}_2 .

Точка $X=G(1,04;3,14)$ из квадрата лежит во внешней (outside) области угла, образованного лучами, исходящими из точки L и направление которых определяют два вектора \vec{u}_1 и \vec{u}_2 .



Исходные данные

В первой строке ввода задается значение a . Во второй — значения координат точки L . В третьей — значение координат вектора \vec{u}_1 . В четвертой — значение координат вектора \vec{u}_2 . В пятой — значение координат испытываемой точки X .

Результат

В строке вывода размещается текст «Inside» или «Outside» в зависимости от вычисленного местоположения точки X по отношению к области угла.

Пример (тест)

исходные данные	результат
5 2 1 0.88 1.42 2.04 0.44 4.42 3.58	Inside
5 2 1 0.88 1.42 2.04 0.44 1.04 3.14	Outside