

1. Число

1 секунда

20 очков

В инженерных и научных расчётах числа зачастую записываются в виде $a \cdot 10^b$, где мантисса a (действительное число, $1 \leq |a| < 10$) указывает значащие цифры а экспонента b (целое число) — порядок числа. Такой способ записи (т.н. стандартный вид) позволяет компактно представлять очень большие и очень маленькие числа (к примеру число 1 200 000 000 000 в стандартном виде записывается как $1,2 \cdot 10^{12}$, а число 0,000 000 000 012 как $1,2 \cdot 10^{-11}$). Человеку, не привыкшему к такому представлению чисел, оно может показаться неудобным, поэтому иногда приходится преобразовывать числа из стандартного вида в десятичную дробь. Необходимо написать программу, выполняющую такое преобразование.

Входные данные. В текстовом файле `ARV.SIS` задано число x в стандартном виде. На первой строке записана мантисса a (до 200 десятичных цифр) и на второй — экспонента b (целое число, $|b| \leq 200$). На третьей строке задана требуемая точность десятичного представления k (целое, $|k| \leq 200$).

Выходные данные. В единственной строке текстового файла `ARV.VAL` вывести число x , округлённое до точности 10^k , десятичном виде. В результате вывести все разряды, оставшиеся после округления, даже если последние разряды содержат нули. Между целой и дробной частями вывести запятую, не точку. Если дробной части нет, запятую не выводить.

Пример.	<code>ARV.SIS</code>	<code>ARV.VAL</code>
	1,2345	1230
	3	
	1	

Округлив число $1,2345 \cdot 10^3 = 1234,5$ до точности $10^1 = 10$ (т.е. до десятков) получим 1230.

Пример.	<code>ARV.SIS</code>	<code>ARV.VAL</code>
	1,234	1234,00
	3	
	-2	

«Округлив» число $1,234 \cdot 10^3 = 1234$ до точности $10^{-2} = 0,01$ (т.е. до сотых) получим 1234,00.

2. Многоугольник

1 секунда

30 очков

Даны координаты центра и одной вершины правильного N -угольника. Написать программу для нахождения периметра, площади и координат остальных вершин этого многоугольника.

Входные данные. На первой строке текстового файла `HN.SIS` дано число вершин многоугольника N ($3 \leq N \leq 20$), на второй строке — координаты центра и на третьей строке — координаты одной из вершин. Все координаты заданы парами действительных чисел, абсолютное значение которых не превышает 100.

Выходные данные. В текстовый файл `HN.VAL` вывести ровно $N + 2$ строк. В первой строке вывести периметр многоугольника, во второй — его площадь, а на следующие N строки — координаты всех вершин многоугольника (по два разделённых пробелом числа на строку). Вершины должны быть выведены в порядке движения по часовой стрелке начиная с вершины, заданной во входном файле. Числа округлять до трёх знаков после запятой.

Пример.	<code>HN.SIS</code>	<code>HN.VAL</code>
	4	8.000
	0.0 0.0	4.000
	1.0 1.0	1.000 1.000
		1.000 -1.000
		-1.000 -1.000
		-1.000 1.000

Это правильный четырёхугольник, т.е. квадрат с длиной стороны 2.

Оценивание. В этом задании за нахождение периметра и площади даётся $1/3$ очков, а за нахождение координат вершин — $2/3$. В случае частичного решения вместо отсутствующих чисел вывести -1 (а не оставлять отсутствующие данные невыведенными!)

3. Драгоценные камни

1 секунда

50 очков

Внутри одного из ледяных пластов Антарктики замёрзло N драгоценных камней. Все камни лежат вдоль одного меридиана (линия с севера на юг). Для того чтобы достать камни, выкапывается траншея с севера на юг. Восточный и западный края траншеи вертикальны, северный и южный делаются под углом 45 градусов к поверхности льда. Стоимость выкапывания траншеи пропорциональна площади её сечения вертикальной плоскостью с севера на юг (один квадратный метр в сечении траншеи обходится в одну крону). При выкапывании нескольких траншей имеющих общие участки, такие участки выкапываются только один раз (соответственно деньги тратятся только один раз).

Прибыль, которую можно получить выкопав драгоценные камни, — это, естественно, стоимость камней минус затраты на их добычу (т.е. на раскопки траншей). Необходимо написать программу для определения того, какие камни нужно выкопать чтобы получить максимальную прибыль.

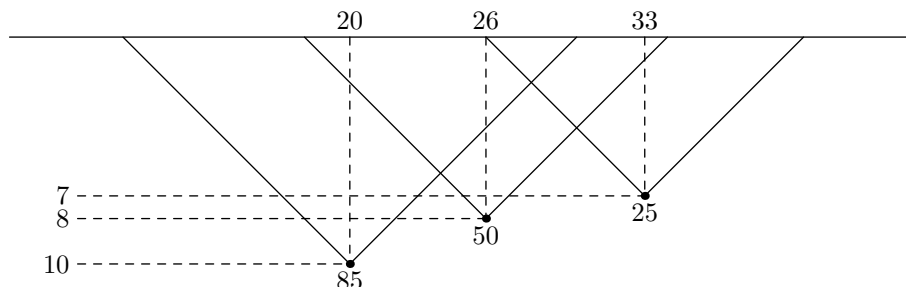
Входные данные. На первой строке текстового файла `KK.SIS` дано число драгоценных камней N ($1 \leq N \leq 1000$). На каждой из следующих N строк даны 3 разделённых пробелом целых числа — данные об одном из камней в виде $X_i Y_i H_i$, где X_i ($0 \leq X_i \leq 10000$) — расстояние (вдоль поверхности льда) от камня до южного полюса (в метрах), Y_i ($0 \leq Y_i \leq 10000$) — глубина залегания камня от поверхности льда (в метрах) и H_i ($0 \leq H_i \leq 10000$) — цена камня в кронах. Камни пронумерованы от 1 до N согласно порядку, в котором они даны в файле. Кривизной земной поверхности можно пренебречь.

Выходные данные. На первой строке текстового файла `KK.VAL` вывести число камней K , которые необходимо выкопать и на следующей строке — K чисел, разделённых пробелами: номера выбранных камней в порядке возрастания. Если существует несколько способов получить наибольшую прибыль, вывести любой.

Пример.

<code>KK.SIS</code>	<code>KK.VAL</code>
3	2
20 10 85	1 3
33 7 25	
26 8 50	

Следующий рисунок иллюстрирует расположение камней и указывает для каждого камня размер траншеи, которую необходимо выкопать для того чтобы его достать. Стоимость выкапывания траншей для извлечения выбранных камней в приведенном примере равна площади объединения двух треугольников слева, которая равна 128. Сумма стоимостей этих камней — 135 крон. Следовательно получаемая прибыль — 7 крон.



Для того, чтобы достать все 3 камня пришлось бы потратить 161 крону, тогда как общая стоимость камней всего 160 крон. Поэтому выкапывание всех 3 камней не принесло бы убытки, а не прибыль.