

Внимание!

В директории **A** лежат тесты (исходные файлы) к задаче **A** (**a1.in**, ..., **a5.in**), в директории **B** – тесты к задаче **B** (**b1.in**, ..., **b5.in**). Вы должны получить ответ на вопрос задачи для **каждого** из исходных файлов.

• Задача A. Многоугольник

Выпуклый многоугольник задан координатами своих вершин. Найдите длину самой длинной диагонали (если их несколько, то длину одной из них).

Входной файл содержит число **N** - количество вершин многоугольника ($4 \leq N \leq 500$) и матрицу $N \times 2$, в каждой строке которой в порядке обхода (например, по часовой стрелке) записаны координаты (**x**, **y**) вершины многоугольника. Ваша программа должна выводить на экран (или в выходной файл) одно число - длину самой длинной диагонали с точностью до 6 знаков после запятой.

Пример:

a.in	на экране или в выходном файле a.out
4	
0 0	5.656854
0 4	
4 4	
4 0	

• Задача B. Покупка билетов

За билетами на премьеру нового мюзикла выстроилась очередь из **N** человек, каждый из которых хочет купить 1 билет. На всю очередь работала только одна касса, поэтому продажа билетов шла очень медленно, приводя «постояльцев» очереди в отчаяние. Самые сообразительные быстро заметили, что, как правило, несколько билетов в одни руки кассир продаёт быстрее, чем когда эти же билеты продаются по одному. Поэтому они предложили нескольким подряд, стоящим людям отдавать деньги первому из них, чтобы он купил билеты на всех.

Однако для борьбы со спекулянтами кассир продавала не более 3-х билетов в одни руки, поэтому договориться, таким образом, между собой могли лишь 2 или 3 подряд стоящих человека.

Известно, что на продажу *i*-му человеку из очереди одного билета кассир тратит A_i секунд, на продажу двух билетов — B_i секунд, трех билетов — C_i секунд. Напишите программу, которая подсчитает минимальное время, за которое могли быть обслужены все покупатели.

Обратите внимание, что билеты на группу объединившихся людей всегда покупает первый из них. Также никто не покупает лишних билетов (есть билетов, которые никому не нужны).

Во входном файле записано сначала число **N**— количество покупателей в очереди ($1 \leq N \leq 5000$). Далее идет **N** троек натуральных чисел A_i , B_i , C_i . Каждое из этих чисел не превышает 3600. Люди в очереди нумеруются, начиная от кассы.

Ваша программа должна выводить на экран (или в выходной файл) одно число минимальное время в секундах, за которое могли быть обслужены все покупатели.

Пример:

b.in	на экране или в выходном файле b.out
5	12
5 10 15	
2 10 15	
5 5 5	
20 20 1	
20 1 1	
2	4
3 4 5	
1 1 1	

• **Задача С. "Змейка"**

Закрасьте как можно больше клеток доски $N*N$ следующим способом:

1. Сначала закрашивается произвольная клетка.
2. Каждая следующая закрашенная клетка должна быть соседней с предыдущей, но не быть соседней с остальными закрашенными клетками. (Соседними считаются клетки, имеющие общую сторону. Клетки, имеющие общую вершину, но не сторону, соседними не считаются).

Ваша программа должна находить одну из оптимальных раскрасок (таких, где количество закрашенных клеток максимально) для случаев а) $N=7$ и б) $N=9$ и выводить ее на экран в виде матрицы $N*N$, в которой закрашенная клетка обозначается 1, а не закрашенная - 0.

Пример:

$N = 4$

1	0	1	1
1	0	0	1
1	0	1	1
1	1	1	0