

Е. Амбициозная улитка

Ограничение времени	5 секунд
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Домашний питомец мальчика Васи — улитка Петя. Петя обитает на бесконечном в обе стороны вертикальном столбе, который для удобства можно представить как числовую прямую. Изначально Петя находится в точке 0.

Вася кормит Петю ягодами. У него есть n ягод, каждая в единственном экземпляре. Вася знает, что если утром он даст Пете ягоду с номером i , то поев и набравшись сил, за остаток дня Петя поднимется на a_i единиц вверх по столбу, но при этом за ночь, потяжелев, съедет на b_i единиц вниз. Параметры различных ягод могут совпадать.

Пете стало интересно, а как оно там, наверху, и Вася взялся ему в этом помочь. Ближайшие n дней он будет кормить Петю ягодами из своего запаса таким образом, чтобы максимальная высота, на которой побывал Петя за эти n дней была максимальной. К сожалению, Вася не умеет программировать, поэтому он попросил вас о помощи. Найдите, максимальную высоту, на которой Петя сможет побывать за эти n дней и в каком порядке Вася должен давать Пете ягоды, чтобы Петя смог её достичь!

Формат ввода

В первой строке входных данных дано число n ($1 \leq n \leq 5 \cdot 10^5$) — количество ягод у Васи. В последующих n строках описываются параметры каждой ягоды. В $i + 1$ строке дано два числа a_i и b_i ($0 \leq a_i, b_i \leq 10^9$) — то, насколько поднимется улитка за день после того, как съест i ягоду и насколько опуститься за ночь.

Формат вывода

В первой строке выходных данных выведите единственное число — максимальную высоту, которую сможет достичь Петя, если Вася будет его кормить оптимальным образом. В следующей строке выведите n различных целых чисел от 1 до n — порядок, в котором Вася должен кормить Петю (i число в строке соответствует номеру ягоды, которую Вася должен дать Пете в i день чтобы Петя смог достичь максимальной высоты).

Пример 1

Ввод



Вывод



3

10

1 5

2 3 1

8 2

4 4

Пример 2

Ввод



Вывод



2

10

7 6

2 1

7 4

Примечания

Во втором примере изначально улитка находится на высоте 0. Пусть сначала Петя накормит её второй ягодой, а затем первой. После того как она съест вторую ягоду, за день она поднимется на 7 (и окажется на высоте 7), а за ночь опустится на 4 (и окажется на высоте 3). После того как она съест первую ягоду, за день она поднимется на 7 (и окажется на высоте 10), а за ночь опустится на 6 (и окажется на высоте 4).

Таким образом, максимальная высота, на которой побывает улитка при данном порядке кормления, равна 10. Нетрудно видеть, что если Петя накормит улитку сначала первой ягодой, а затем второй, то максимальная высота, на которой побывает улитка, будет меньше.

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы	
18 мар 2024, 12:05:39	110013517	E	C++20 (GCC 14.1)	OK	-	200ms	12.64Mb	-	-	отчёт
18 мар 2024, 11:55:36	110012731	E	C++20 (GCC 14.1)	WA	-	2ms	292.00Kb	4	-	отчёт
18 мар 2024, 11:44:37	110012031	E	C++20 (GCC 14.1)	WA	-	2ms	296.00Kb	4	-	отчёт
18 мар 2024, 11:41:04	110011809	E	C++20 (GCC 14.1)	WA	-	2ms	292.00Kb	1	-	отчёт
17 мар 2024, 21:51:57	109981665	E	C++20 (GCC 14.1)	WA	-	2ms	296.00Kb	16	-	отчёт
17 мар 2024, 21:49:52	109981379	E	C++20 (GCC 14.1)	WA	-	2ms	296.00Kb	16	-	отчёт
17 мар 2024, 21:45:38	109980803	E	C++20 (GCC 14.1)	WA	-	2ms	296.00Kb	15	-	отчёт
17 мар 2024, 21:43:19	109980508	E	C++20 (GCC 14.1)	WA	-	2ms	296.00Kb	15	-	отчёт
17 мар 2024, 20:31:02	109971334	E	C++20 (GCC 14.1)	WA	-	2ms	292.00Kb	7	-	отчёт
17 мар 2024, 16:51:51	109946015	E	C++20 (GCC 14.1)	WA	-	2ms	292.00Kb	4	-	отчёт

№	Вердикт	Ресурсы	Баллы
1	ok	2ms / 296.00Kb	-
2	ok	2ms / 296.00Kb	-
3	ok	2ms / 296.00Kb	-
4	ok	2ms / 288.00Kb	-
5	ok	2ms / 296.00Kb	-
6	ok	2ms / 292.00Kb	-
7	ok	2ms / 292.00Kb	-
8	ok	2ms / 296.00Kb	-
9	ok	2ms / 292.00Kb	-
10	ok	2ms / 292.00Kb	-
11	ok	2ms / 296.00Kb	-
12	ok	2ms / 296.00Kb	-
13	ok	2ms / 292.00Kb	-
14	ok	2ms / 292.00Kb	-
15	ok	2ms / 288.00Kb	-
16	ok	2ms / 224.00Kb	-
17	ok	2ms / 228.00Kb	-
18	ok	2ms / 228.00Kb	-
19	ok	5ms / 464.00Kb	-
20	ok	36ms / 2.69Mb	-
21	ok	36ms / 2.59Mb	-
22	ok	35ms / 2.61Mb	-
23	ok	35ms / 2.56Mb	-
24	ok	35ms / 2.73Mb	-
25	ok	36ms / 2.74Mb	-
26	ok	70ms / 5.02Mb	-
27	ok	70ms / 4.90Mb	-

26	ok	70ms / 5.02Mb	-
27	ok	70ms / 4.90Mb	-
28	ok	72ms / 4.90Mb	-
29	ok	69ms / 4.90Mb	-
30	ok	180ms / 12.14Mb	-
31	ok	186ms / 12.64Mb	-
32	ok	200ms / 11.76Mb	-
33	ok	174ms / 11.76Mb	-
34	ok	179ms / 11.76Mb	-
35	ok	178ms / 11.87Mb	-
36	ok	182ms / 11.75Mb	-
37	ok	179ms / 11.77Mb	-
38	ok	183ms / 12.64Mb	-