Задача Jaguar. Выход участников

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

На приоткрытую олимпиаду школьников по программированию пришло n участников, пронумерованных числами от 1 до n. Участник под номером i одет в футболку цвета a_i. Организаторы соревнований собираются приглашать участников в зал соревнований по очереди. Чтобы этот процесс был приятен глазу, они хотят избежать ситуаций, когда два участника подряд выходят в футболках одинакового цвета. Для этого участники будут приглашаться по следующему алгоритму:

- Первым в зал соревнований приглашается участник под номером 1.
- Далее каждый раз в зал приглашается участник, у которого цвет футболки отличается от цвета футболки предыдущего вошедшего участника. Если таких участников несколько, то выбирается участник с наименьшим номером.
- Наконец, если у всех оставшихся участников цвет футболки совпадает с последним вошедшим участником, то все оставшиеся участники приглашаются в порядке увеличения их номеров.

В ночь перед олимпиадой организаторы подготовили план выхода участников, но прямо перед открытием они заметили, что участники, имеющие соседние номера, иногда обмениваются футболками. Конечно, это привело к тому, что предыдущий план перестал соответствовать правилам, и организаторам потребовалась разработка нового.

Требуется отвечать на запросы двух типов:

1. Два участника с номерами x_i и $(x_i + 1)$ меняются футболками.
2. Найти, каким по счету выйдет участник под номером y_i, если участники начнут выходить по описанному выше алгоритму, начиная с первого с учетом всех предыдущих обменов футболками.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leqslant n \leqslant 500\,000$, $1 \leqslant q \leqslant 500\,000$) — количество участников на соревновании и количество запросов.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$) — изначальные цвета футболок участников в порядке их нумерации.

В следующих q строках описываются запросы. В i-й из них в начале содержится целое число t_i ($1 \leqslant t_i \leqslant 2$) — тип i-го запроса.

1. Если $t_i = 1$, то далее строка содержит одно целое число x_i ($1 \leqslant x_i < n$). В этом случае в i-м запросе участники с номерами x_i и $(x_i + 1)$ меняются футболками.
2. Если $t_i = 2$, то далее строка содержит одно целое число y_i ($1 \leqslant y_i \leqslant n$). В этом случае в i-м запросе необходимо найти, каким по счету выйдет участник под номером y_i, если участники начнут выходить по описанному выше алгоритму, начиная с первого с учетом всех предыдущих обменов футболками.

Формат выходных данных

Для каждого запроса второго типа в отдельной строке выведите одно целое число — ответ на запрос.

Гарантируется, что существует хотя-бы один запрос второго типа.

Примеры

ввод	вывод
10 10 3 1 1 2 2 1 1 2 2 2 2 2 2 3 2 4 2 10 1 1 2 2 2 3 2 4 2 5 2 10	2 4 3 10 2 3 4 6 10
10 10 1 2 2 2 3 4 5 6 7 8 2 1 1 1 2 2 1 2 2 3 1 3 2 4 1 4 2 3 2 5	1 2 2 2 5 4

Пояснение

В первом примере к задаче в изначальной конфигурации участники выходят в порядке:

$$1,\quad 2,\quad 4,\quad 3,\quad 5,\quad 6,\quad 8,\quad 7,\quad 9,\quad 10$$

То есть участник с номером 2 выходит вторым, участник с номером 3 – четвёртым, участник с номером 4 – третьим, а участник с номером 10 – десятым.

После того, как участники с номерами 1 и 2 обменялись футболками, футболки участников имеют следующие цвета:

$$1,\quad 3,\quad 1,\quad 2,\quad 2,\quad 1,\quad 1,\quad 2,\quad 2,\quad 2$$

Поэтому после данного изменения участники будут выходить в порядке:

$$1,\quad 2,\quad 3,\quad 4,\quad 6,\quad 5,\quad 7,\quad 8,\quad 9,\quad 10$$

То есть участник с номером 2 выйдет вторым, участник с номером 3 – третьим, участник с номером 4 – четвёртым, участник с номером 5 – шестым, а участник с номером 10 – десятым.

Во втором примере к задаче в изначальной конфигурации участники выходят в порядке:

$$1,\quad 2,\quad 5,\quad 3,\quad 6,\quad 4,\quad 7,\quad 8,\quad 9,\quad 10$$

То есть участник с номером 1 выходит первым.

После того, как участники с номерами 1 и 2 обменялись футболками, футболки участников имеют следующие цвета:

$$2,\quad 1,\quad 2,\quad 2,\quad 3,\quad 4,\quad 5,\quad 6,\quad 7,\quad 8$$

Поэтому после данного изменения участники будут выходить в порядке:

$$1,\quad 2,\quad 3,\quad 5,\quad 4,\quad 6,\quad 7,\quad 8,\quad 9,\quad 10$$

То есть участник с номером 2 выходит вторым.

После того, как участники с номерами 2 и 3 обменялись футболками, футболки участников имеют следующие цвета:

$$2,\quad 2,\quad 1,\quad 2,\quad 3,\quad 4,\quad 5,\quad 6,\quad 7,\quad 8$$

Поэтому после данного изменения участники будут выходить в порядке:

$$1,\quad 3,\quad 2,\quad 5,\quad 4,\quad 6,\quad 7,\quad 8,\quad 9,\quad 10$$

То есть участник с номером 3 выходит вторым.

После того, как участники с номерами 3 и 4 обменялись футболками, футболки участников имеют следующие цвета:

$$2,\quad 2,\quad 2,\quad 1,\quad 3,\quad 4,\quad 5,\quad 6,\quad 7,\quad 8$$

Поэтому после данного изменения участники будут выходить в порядке:

$$1,\quad 4,\quad 2,\quad 5,\quad 3,\quad 6,\quad 7,\quad 8,\quad 9,\quad 10$$

То есть участник с номером 4 выходит вторым.

После того, как участники с номерами 4 и 5 обменялись футболками, футболки участников имеют следующие цвета:

$$2,\quad 2,\quad 2,\quad 3,\quad 1,\quad 4,\quad 5,\quad 6,\quad 7,\quad 8$$

Поэтому после данного изменения участники будут выходить в порядке:

$$1,\quad 4,\quad 2,\quad 5,\quad 3,\quad 6,\quad 7,\quad 8,\quad 9,\quad 10$$

То есть участник с номером 3 выходит пятым, а участник с номером 5 – четвёртым.

Система оценки

Тесты к этой задаче состоят из 12 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		n, q		
0	0	–	–	Тесты из условия.
1	7	$n, q \leqslant 500$	0	
2	9	$n, q \leqslant 5\,000$	0, 1	
3	5	$n, q \leqslant 10\,000$	0 – 2	
4	10	$n, q \leqslant 100\,000$	0 – 3	
5	8	$n, q \leqslant 200\,000$	0 – 4	
6	7	$n, q \leqslant 300\,000$	0 – 5	
7	9	–	–	$1 \leqslant a_i \leqslant 2$
8	9	–	7	$1 \leqslant a_i \leqslant 5$
9	11	–	–	Для любых $i \neq j$: $a_i = 1$ или $a_i \neq a_j$ Если $t_i = 2$, то $y_i = \lceil \frac{9n}{10} \rceil$
10	8	–	9	Для любых $i \neq j$: $a_i = 1$ или $a_i \neq a_j$
11	9	–	9	Если $t_i = 2$, то $y_i = \lceil \frac{9n}{10} \rceil$
12	8	–	0 – 11	**Offline-проверка.**

Задача Volvo. Перестановки и запросы

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Вам дана перестановка p длины n. Перестановка длины n — это массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. *Стоимостью* перестановки определим сумму по всем i от 1 до n величины $(p_i)^i$ (i-й элемент перестановки возведенный в степень i). Таким образом, *стоимость* перестановки p равна

$$\sum_{i=1}^{n} (p_i)^i$$

Поступает q запросов трех типов:

1. Развернуть. После этого ваша перестановка p заменяется на перестановку q, такую, что $q_i = p_{n-i+1}$ для всех i от 1 до n.
2. Перевернуть. После этого ваша перестановка p заменяется на перестановку q, такую, что $q_i = n - p_i + 1$ для всех i от 1 до n.
3. Взять обратную. После этого ваша перестановка p заменяется на перестановку q, такую, что $q_{p_i} = i$ для всех i от 1 до n.

Обратите внимание, что после каждой операции p остаётся перестановкой.

После каждого запроса вам нужно вывести *стоимость* перестановки.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leqslant n, q \leqslant 100\,000$) — длина перестановки и количество запросов.

Вторая строка содержит n целых положительных чисел $p_1, p_2, \ldots, p_n$ ($1 \leqslant p_i \leqslant n$) — элементы перестановки. Гарантируется, что все p_i различны.

Третья строка содержит q целых положительных чисел $b_1, b_2, \ldots, b_q$ ($1 \leqslant b_i \leqslant 3$) — описание запросов. Число b_i означает, что i-й запрос изменения, который нужно применить к перестановке, имеет тип b_i.

Формат выходных данных

Выведите q чисел, i-е из которых — остаток *стоимости* перестановки по модулю $998\,244\,353$, после применения первых i запросов.

Примеры

ввод	вывод
5 5 1 2 3 4 5 1 2 3 1 2	65 3413 3413 65 3413
5 6 5 3 1 4 2 3 3 1 2 3 1	293 303 3225 215 317 3209

Пояснение

Разберем второй пример.

Изначально $p = [5, 3, 1, 4, 2]$.

Первый запрос имеет тип 3, то есть взять обратную. Перестановка после этого запроса превращается в $[3, 5, 2, 4, 1]$. *Стоимостью* этой перестановки является $3^1 + 5^2 + 2^3 + 4^4 + 1^5 = 3 + 25 + 8 + 256 + 1 = 293$.

Второй запрос имеет тип 3, то есть взять обратную. Перестановка после этого запроса превращается в $[5, 3, 1, 4, 2]$. *Стоимостью* этой перестановки является $5^1 + 3^2 + 1^3 + 4^4 + 2^5 = 5 + 9 + 1 + 256 + 32 = 303$.

Третий запрос имеет тип 1, то есть развернуть. Перестановка после этого запроса превращается в $[2, 4, 1, 3, 5]$. *Стоимостью* этой перестановки является $2^1 + 4^2 + 1^3 + 3^4 + 5^5 = 3225$.

Четвертый запрос имеет тип 2, то есть перевернуть. Перестановка после этого запроса превращается в $[4, 2, 5, 3, 1]$. *Стоимостью* этой перестановки является $4^1 + 2^2 + 5^3 + 3^4 + 1^5 = 215$.

Пятый запрос имеет тип 3, то есть взять обратную. Перестановка после этого запроса превращается в $[5, 2, 4, 1, 3]$. *Стоимостью* этой перестановки является $5^1 + 2^2 + 4^3 + 1^4 + 3^5 = 317$.

Последний запрос имеет тип 1, то есть развернуть. Перестановка после этого запроса превращается в $[3, 1, 4, 2, 5]$. *Стоимостью* этой перестановки является $3^1 + 1^2 + 4^3 + 2^4 + 5^5 = 3209$.

Система оценки

Тесты к этой задаче состоят из 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	q		
0	0	–	–	–	Тесты из условия.
1	15	$n \leqslant 1000$	$q \leqslant 1000$	0	
2	22	–	–	–	$b_i = b_j$ для всех $1 \leqslant i, j \leqslant q$
3	26	–	–	–	$b_i \leqslant 2$ для всех $1 \leqslant i \leqslant q$
4	16	–	–	–	$p_i = i$ для всех $1 \leqslant i \leqslant n$
5	21	–	–	0 – 4	

Задача Toyota. Задачи от Саши

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Саша недавно переехал в многоэтажный дом. Всего в доме есть n этажей, пронумерованных числами от 1 до n. На каждом этаже дома живет ровно один жилец. Между этажами построена $n-1$ лестница, при этом лестницы не обязательно построены между соседними этажами. Известно, что для каждого этажа, кроме первого, есть ровно одна лестница, ведущая на более нижний этаж. Для i-го этажа ($2 \leqslant i \leqslant n$) такая лестница ведет на этаж с номером p_i.

Саша собирается решить k задач, пронумерованных от 1 до k. Для задачи с номером i Саша вычислил, что оптимальнее всего ее решать на этаже с номером x_i. Так как задачи отличаются между собой, то все значения x_i различны.

Решать задачи одному скучно, поэтому для каждой задачи Саша хочет позвать хотя-бы одного жильца, чтобы решать ее вместе. Однако жильцы дома очень не любят подниматься по лестницам и готовы только спускаться на этажи, на которых будут решаться задачи. Поэтому Саша может позвать жильца с этажа j решать задачу i только если с j-го этажа есть возможность добраться до этажа x_i, используя несколько, возможно ноль, лестниц, ведущих каждый раз на этаж с меньшим номером. Таким образом, жилец этажа j может решать задачу i только если $j = x_i$ или $p_j = x_i$, или $p_{p_j} = x_i$ и т.д.

Жильцы очень не любят лишний раз спускаться по лестнице. Поэтому если Саша позовет некоторое множество людей решать задачу, они будут готовы собраться решать задачу только **на самом высоком этаже, до которого они все смогут спуститься**. Например, если с этажа 3 ведет лестница на этаж 2, то Саша не сможет позвать жильцов этажей 2 и 3 решать задачу на этаже 1, так как все жильцы могут собраться на более высоком этаже.

Саша не любит казаться навязчивым, поэтому с каждого этажа Саша позовет жильца решать не более одной задачи. При этом с некоторых этажей Саша может не звать жильцов решать задачи.

У Саши есть ещё одна любимая задача, которой он не станет делиться ни с кем, кроме вас. Но чтобы он вам ее рассказал, надо помочь ему посчитать количество различных способов позвать жильцов решать с ним задачи, чтобы все ограничения были выполнены. Два способа считаются различными, если хотя-бы одну задачу решает разное множество жильцов.

Формат входных данных

Первая строка содержит два целых числа n и k ($3 \leqslant n \leqslant 10^6$, $1 \leqslant k \leqslant \min(n, 2000)$) — количество этажей в доме и количество задач.

Вторая строка содержит k целых чисел x_1, x_2, $\ldots$, x_k ($1 \leqslant x_i \leqslant n$) — этажи, на которых Саша будет решать задачи. Гарантируется, что все x_i различны.

Третья строка содержит $n-1$ целое число p_2, p_3, $\ldots$, p_n ($1 \leqslant p_i < i$), где p_i описывает номер этажа, на который ведет лестница вниз с этажа i.

Формат выходных данных

Выведите одно число — количество различных способов позвать жильцов дома решать задачи вместе с Сашей по модулю 998 244 353.

Примеры

ввод	вывод
3 1 1 1 1	5
6 2 2 5 1 2 3 4 5	12
7 3 2 7 1 1 1 2 2 3 3	62

Пояснение

В первом примере у Саши есть пять способов позвать решать задачи жильцов дома:

- только с жильцом на этаже 1;
- с жильцами на этажах 1 и 2;
- с жильцами на этажах 1 и 3;
- с жильцами на этажах 1, 2 и 3;
- с жильцами на этажах 2 и 3;

Позвать решать задачу только жильца с этажа 2 Саша не может, так как тогда самым высоким этажом, на котором смогут собраться все жильцы, желающие решать задачу, будет этаж 2, а Саша хочет решать задачу на этаже 1.

Во втором примере двумя различными подходящими способами позвать жильцов решать задачи могут быть следующие:

- Позвать жильцов с этажей 2 и 6 решать первую задачу, а жильца с этажа 5 решать вторую задачу.
- Позвать жильца с этажа 2 решать первую задачу, а жильцов с этажей 5 и 6 решать вторую задачу.

Система оценки

Тесты к этой задаче состоят из 9 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	k		
0	0	–	–	–	Тесты из условия.
1	12	$n \leqslant 10$	$k \leqslant 10$	0	
2	13	$n \leqslant 500$	$k \leqslant 500$	0, 1	
3	9	–	$k = 1$	–	
4	10	–	–	–	$p_i = i - 1$
5	13	–	–	4	Каждый этаж соединён не более чем с двумя этажами с большим номером
6	14	$n \leqslant 200\,000$	$k \leqslant 500$	0 – 2	
7	11	–	$k \leqslant 500$	0 – 3, 6	
8	10	–	$k \leqslant 1000$	0 – 3, 6, 7	
9	8	–	–	0 – 8	**Offline-проверка**

Задача Mercedes. Волнение перед олимпиадой

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Перед входом на закрытую олимпиаду собралось n участников, пронумерованных от 1 до n. Известно, что каждую минуту в зал соревнования будут запускать по одному участнику в порядке возрастания их номеров. Через минуту в зал соревнований зайдёт первый участник, через две минуты — второй участник и так далее. Таким образом, i-й участник зайдет в зал соревнования через i минут после начала запуска участников в зал соревнований.

У каждого участника есть некоторый уровень волнения перед олимпиадой. Уровень волнения каждого участника задается некоторым целым (возможно отрицательным) числом. До начала запуска участников, уровень волнения участника с номером i равен a_i. Каждую минуту уровень волнения участника будет изменяться на величину b_i. Таким образом, через x минут после начала запуска участников уровень волнения участника с номером i будет равен $a_i + x \cdot b_i$.

Александр — опытный психолог, который решил успокоить участников в очереди на вход на олимпиаду. Для этого Александр говорит с участниками и успокаивает их. С каждым участником Александр может поговорить не более одного раза. После общения с Александром уровень волнения участника становится равным 0 и перестает меняться после этого. *Эффективностью* работы Александра с участником с номером i считается величина волнения участника в момент общения с Александром. Таким образом, если Александр пообщается с участником i через t_i минут после начала запуска участников на олимпиаду, *эффективность* будет равна $a_i + t_i \cdot b_i$. Обратите внимание, что если уровень волнения участника был отрицательным, то и *эффективность* работы будет отрицательной.

Александр будет работать с участниками в порядке возрастания номеров. Однако Александру не обязательно говорить со всеми участниками, то есть возможно, что Александр не будет общаться с последними участниками в очереди. Обратите внимание, что с каждым участником Александр будет общаться не более одного раза и случиться это должно до момента входа участника в зал соревнования. При этом в каждую минуту Александр может общаться сразу с несколькими участниками. Более формально, процесс работы Александра можно описать следующим образом:

- Всего Александр пообщается с первыми k участниками в очереди, где k он выбирает сам.
- Для каждого из первых k участников зафиксируем целое неотрицательное число t_i — время общения с Александром. Обратите внимание, что t_i может быть равно нулю, что означает, что Александр поговорил с i-м участником до того, как в зал соревнований пустили первого участника.
- Для каждого i от 1 до k, $t_i < i$, так как Александр должен общаться с участниками до их входа в зал соревнования.
- Для каждого i от 1 до $k - 1$, $t_i \leqslant t_{i+1}$, так как Александр общается с участниками в порядке возрастания их номеров.
- *Суммарная эффективность* общения Александра с участниками будет описываться следующей формулой:

$$\sum_{i=1}^{k} (a_i + t_i \cdot b_i)$$

Для Александра заранее зафиксирован план работы с участниками. План работы с участниками задается последовательностью из n целых чисел p_1, p_2, $\ldots$, p_n. Для каждого i, если $p_i \neq -1$, то к моменту запуска первых i участников в зал соревнования (через i минут после начала запуска участников), Александр должен поговорить с первыми p_i участниками и ни с кем больше. В этом случае также гарантируется, что $p_i \geqslant i$. Если же $p_i = -1$, то это означает, что нет никаких ограничений на число участников, с которыми Александр должен поработать спустя i минут после начала запуска участников.

Более формально, если $p_i \neq -1$, то:

- $p_i \geqslant i$;
- $t_{p_i} < i$;
- Для любого j, такого что $p_i < j \leqslant k$ верно, что $t_j \geqslant i$.

Помогите Александру определить, какой может быть максимальная величина *суммарной эффективности* работы с участниками при выполнении всех ограничений. Гарантируется, что ответ всегда существует.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leqslant n \leqslant 10^6$) — количество участников, стоящих в очереди на вход в зал соревнований олимпиады.

Следующие n строк содержат по два целых числа a_i и b_i ($-10^9 \leqslant a_i \leqslant 10^9$, $-10^6 \leqslant b_i \leqslant 10^6$) — параметры волнения i-го участника.

Следующая строка содержит n целых чисел $p_1, p_2, \ldots, p_n$ ($i \leqslant p_i \leqslant n$ или $p_i = -1$) — описание плана работы для Александра.

Гарантируется, что для любой пары $1 \leqslant i < j \leqslant n$ выполнено $p_i \leqslant p_j$, если $p_i \neq -1$ и $p_j \neq -1$.

Формат выходных данных

Выведите одно число — максимальную возможную *суммарную эффективность* работы Александра.

Можно показать, что Александр всегда сможет выполнить свою работу, соблюдая все дополнительные ограничения и план работ.

Примеры

ввод	вывод
4 3 -6 4 -10 -7 -3 -3 6 3 3 -1 -1	15
4 -6 -1 -5 14 0 10 -30 2 2 3 -1 -1	-1
4 -6 -1 -5 14 0 10 -30 2 -1 -1 -1 -1	23

Пояснение

В первом тестовом примере оптимально выбрать $k = 4$ и $t = \{0, 0, 0, 3\}$. Тогда *суммарная эффективность* будет равна:

$$(3 + t_0 \cdot (-6)) + (4 + t_1 \cdot (-3)) + (-7 + t_2 \cdot (-3)) + (-3 + t_4 \cdot 6) =$$

$$= (3 + 0 \cdot (-6)) + (4 + 0 \cdot (-3)) + (-7 + 0 \cdot (-3)) + (-3 + 3 \cdot 6) = 15$$

Во втором тестовом примере оптимально выбрать $k = 3$ и $t = \{0, 0, 1\}$. Тогда *суммарная эффективность* будет равна:

$$(-6 + t_0 \cdot (-1)) + (-5 + t_1 \cdot 14) + (0 + t_2 \cdot 10) = (-6 + 0 \cdot (-1)) + (-5 + 0 \cdot 14) + (0 + 1 \cdot 10) = -1$$

В третьем тестовом примере оптимально выбрать $k = 3$ и $t = \{0, 1, 2\}$. Тогда *суммарная эффективность* будет равна:

$$(-6 + t_0 \cdot (-1)) + (-5 + t_1 \cdot 14) + (0 + t_2 \cdot 10) = (-6 + 0 \cdot (-1)) + (-5 + 1 \cdot 14) + (0 + 2 \cdot 10) = 23$$

Система оценки

Тесты к этой задаче состоят из 14 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

<table>
<tr><th rowspan="2">Группа</th><th rowspan="2">Баллы</th><th colspan="3">Ограничения</th><th rowspan="2">Необх. группы</th><th rowspan="2">Комментарий</th></tr>
<tr><th>n</th><th>b_i</th><th>p_i</th></tr>
<tr><td>0</td><td>0</td><td>–</td><td>–</td><td>–</td><td>–</td><td>Тесты из условия.</td></tr>
<tr><td>1</td><td>6</td><td rowspan="2">$n \leqslant 100$</td><td rowspan="2">–</td><td>$p_i = -1$</td><td>–</td><td rowspan="2"></td></tr>
<tr><td>2</td><td>6</td><td>–</td><td>0 – 1</td></tr>
<tr><td>3</td><td>7</td><td rowspan="2">$n \leqslant 5000$</td><td rowspan="2">–</td><td>$p_i = -1$</td><td>1</td><td rowspan="2"></td></tr>
<tr><td>4</td><td>6</td><td>–</td><td>0 – 3</td></tr>
<tr><td>5</td><td>7</td><td rowspan="2">–</td><td rowspan="2">$b_i \leqslant 0$</td><td>$p_i = -1$</td><td>–</td><td rowspan="2"></td></tr>
<tr><td>6</td><td>5</td><td>–</td><td>5</td></tr>
<tr><td>7</td><td>7</td><td rowspan="2">–</td><td rowspan="2">$b_i \geqslant 0$</td><td>$p_i = -1$</td><td>–</td><td rowspan="2"></td></tr>
<tr><td>8</td><td>5</td><td>–</td><td>7</td></tr>
<tr><td>9</td><td>9</td><td rowspan="2">–</td><td rowspan="2">$b_i \leqslant b_{i+1}$</td><td>$p_i = -1$</td><td>–</td><td rowspan="2"></td></tr>
<tr><td>10</td><td>8</td><td>–</td><td>9</td></tr>
<tr><td>11</td><td>10</td><td rowspan="2">$n \leqslant 100\,000$</td><td rowspan="2">–</td><td>$p_i = -1$</td><td>–</td><td rowspan="2">Количество $b_i > 0$ не больше 10</td></tr>
<tr><td>12</td><td>7</td><td>–</td><td>11</td></tr>
<tr><td>13</td><td>9</td><td rowspan="2">–</td><td rowspan="2">–</td><td>$p_i = -1$</td><td>1, 3, 5, 7, 9, 11</td><td rowspan="2">Offline-проверка</td></tr>
<tr><td>14</td><td>8</td><td>–</td><td>0 – 13</td></tr>
</table>