Problem Jaguar. Въвеждане на участници

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	3 seconds
Memory limit:	512 megabytes

На откриването на Откритата ученическа олимпиада по програмиране пристигнали n участници, номерирани с числата от 1 до n. Участникът с номер i е облечен с тениска с цвят a_i. Организаторите на състезанието решили да поканват участниците в залата на състезанието в определен ред. За да може този процес да е приятен за очите, те искат да избегнат ситуацията, когото два последователно поканени участници са с тениски с еднакъв цвят. За целта, участниците ще бъдат поканвани по следния алгоритъм:

- Първи в залата на състезанието се кани участникът с номер 1.
- По-нататък, всеки път в залата се кани участник, който има цвят на тениската, различен от цвета на тениската на предишния участник. Ако такива участници са няколко, се избира участникът с най-малък номер.
- Накрая, ако цветовете на тениските на всички останали състезатели съвпадат с тази на последният поканен, всечки останали участници се канят по ред на номерата.

В нощта преди олимпиадата организаторите подготовили план за влизане на участниците, но непосредствено преди откриването те забелязали, че участниците, със съседни номера, понякога си разменят тениските. Естествено, това довело до там, че предварителният план вече не съответствал на правилата, и организаторите трябвало да разработят нов.

Налага се да се отговаря на заявки от два типа:

1. Двама участници с номера x_i и $(x_i + 1)$ си сменят тениските.
2. Да се намери кой подред ще влезе участник с номер y_i, ако участниците влизат по описания вече алгоритъм, започвайки от първия, но като се отчитат всички смени на тениски до момента.

Input

Първият ред на стандартния вход съдържа две цели числа n и q ($1 \le n \le 500\,000$, $1 \le q \le 500\,000$) — броя участници в състезанието и броя заявки.

Вторият ред съдържа n цели числа a_1, a_2, ..., a_n ($1 \le a_i \le n$) — първоначалните цветове на тениските на участниците по реда на тяхната номерация.

На следващите q реда се описват заявките. В i-тия от тях в началото е записано цяло число t_i ($1 \le t_i \le 2$) — типа на i-тата заявка.

1. Ако $t_i = 1$, редът съдържа още едно цяло число x_i ($1 \le x_i < n$). В този случай при i-тата заявка участниците с номера x_i и $(x_i + 1)$ си сменят тениските.
2. Ако $t_i = 2$, редът съдържа още едно цяло число y_i ($1 \le y_i \le n$). В този случай при i-м заявка трябва да се намери, кой подред ще влезе участникът с номер y_i, ако участниците започнат да влизат по описания алгоритъм, започвайки от първия, но като се отчитат всички смени на тениски до момента.

Output

За всяка заявка от втория тип изведете едно цяло число — отговорът на заявката.

Гарантирано е, че съществува поне една заявка от втория тип.

Examples

input	output
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

Note

В първия пример към задачата според началната конфигурация участниците влизат в следния ред:

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

т.е. участник с номер 2 влиза втори, участник с номер 3 – четвърти, участник с номер 4 – трети, а участник с номер 10 – десети.

След като участниците с номера 1 и 2 си сменят тениските, тениските на участниците имат следните цветове:

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

Затова, след това изменение, участниците ще влизат в следния ред:

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

т.е. участник с номер 2 влеза втори, участник с номер 3 – трети, участник с номер 4 – четвърти, участник с номер 5 – шести, а участник с номер 10 – десети.

Във втория пример към задачата според началната конфигурация участниците влизат в следния ред::

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

т.е. участник с номер 1 влиза първи.

След като участниците с номера 1 и 2 си сменят тениските, тениските на участниците имат следните цветове:

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

Затова, след това изменение, участниците ще влизат в следния ред:

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

т.е. участник с номер 2 влиза втори.

След като участниците с номера 2 и 3 си сменят тениските, тениските на участниците имат следните цветове:

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

Затова, след това изменение, участниците ще влизат в следния ред:

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

т.е. участник с номер 3 влиза втори.

След като участниците с номера 3 и 4 си сменят тениските, тениските на участниците имат следните цветове:

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

Затова, след това изменение, участниците ще влизат в следния ред:

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

т.е. участник с номер 4 влиза втори.

След като участниците с номера 3 и 4 си сменят тениските, тениските на участниците имат следните цветове:

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

Затова, след това изменение, участниците ще влизат в следния ред:

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

т.е. участник с номер 3 влиза пети, а участник с номер 5 – четвърти.

Scoring

Тестовете към тази задача се състоят от 12 групи. Точките за всяка група се дават само ако са преминати всички тестове от групата и всички тестове от някои от предходните групи. Обърнете внимание, преминаването на тестовете от условието не е необходимо за някои от групите. **Offline-проверка** означава, че резултатите от тестването на вашите решения за дадена група ще бъдат достъпни след края на състезанието. Общия брой точки за всяка група е равен на максималния брой точки, получени за тази група тестове от всички събмити.

Група	Точки	Доп. ограничения	Необх. групи	Коментари
		n, q		
0	0	–	–	Тестовете от условието.
1	7	$n, q \leq 500$	0	
2	9	$n, q \leq 5\,000$	0, 1	
3	5	$n, q \leq 10\,000$	0 – 2	
4	10	$n, q \leq 100\,000$	0 – 3	
5	8	$n, q \leq 200\,000$	0 – 4	
6	7	$n, q \leq 300\,000$	0 – 5	
7	9	–	–	$1 \leq a_i \leq 2$
8	9	–	7	$1 \leq a_i \leq 5$
9	11	–	–	За произволни $i \neq j$: $a_i = 1$ или $a_i \neq a_j$ Ако$t_i = 2$, то $y_i = \lceil \frac{9n}{10} \rceil$
10	8	–	9	За произволни $i \neq j$: $a_i = 1$ или $a_i \neq a_j$
11	9	–	9	Ако $t_i = 2$, то $y_i = \lceil \frac{9n}{10} \rceil$
12	8	–	0 – 11	**Offline-проверка.**

Problem Volvo. Пермутации и заявки

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

Дадена е пермутация p с дължина n. Пермутация с дължина n, се нарича масив, който е съставен от n различни цели числа 1 до n в произволен ред. *Стойност* на пермутация ще наричаме сума на величините $(p_i)^i$ (i-тия елемент на пермутацията повдигнат на степен i) за всяко i от 1 до n. Тогава, *стойността* на пермутацията p е равна на

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

Постъпват q заявки от три типа:

1. Разгъни. След тази заявка пермутацията p се заменя с пермутацията q, такава, че $q_i = p_{n-i+1}$ за всички i от 1 до n.
2. Обърни. След тази заявка пермутацията p се заменя с пермутацията q, такава, че $q_i = n - p_i + 1$ за всички i от 1 до n.
3. Вземи обратното. След тази заявка пермутацията p се заменя с пермутацията q, такава, че $q_{p_i} = i$ за всички i от 1 до n.

Обърнете внимание, че след всяка заявка p остава пермутация.

След всяка заявка трябва да се изведе *стойността* на пермутацията.

Input

Първия ред на стандартния вход съдържа две цели числа n и q ($1 \le n, q \le 100\,000$) — дължината на пермутацията и броя заявки.

Втория ред съдържа n цели положителни числа p_1, p_2, ..., p_n ($1 \le p_i \le n$) — елементите на пермутацията. Гарантирано е, че всички p_i са различни.

Третия ред съдържа q цели положителни числа b_1, b_2, ..., b_q ($1 \le b_i \le 3$) — описание на заявките. Числото b_i означава, че за i-тата заявка изменението, което трябва да приложим към пермутацията, е от тип b_i.

Output

Изведете q числа, i-тото, от които е остатъкът от *стойността* на пермутацията по модул 998 244 353, след изпълнение на първите i заявки.

Examples

input	output
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

Note

Да разгледаме втория пример.

В началото $p = [5, 3, 1, 4, 2]$.

Първата заявка е от тип 3, т.е. вземи обратната. След тази заявка пермутацията се превръща в $[3, 5, 2, 4, 1]$. *Стойността* на тази пермутация е $3^1 + 5^2 + 2^3 + 4^4 + 1^5 = 3 + 25 + 8 + 256 + 1 = 293$.

Втората заявка е от тип 3, т.е. вземи обратната. След тази заявка пермутацията се превръща в $[5, 3, 1, 4, 2]$. *Стойността* на тази пермутация е $5^1 + 3^2 + 1^3 + 4^4 + 2^5 = 5 + 9 + 1 + 256 + 32 = 303$.

Третата заявка е от тип 1, т.е. разшири. След тази заявка пермутацията се превръща в $[2, 4, 1, 3, 5]$. *Стойността* на тази пермутация е $2^1 + 4^2 + 1^3 + 3^4 + 5^5 = 3225$.

Четвъртата заявка е от тип 2, т.е. обърни. След тази заявка пермутацията се превръща в $[4, 2, 5, 3, 1]$. *Стойността* на тази пермутация е $4^1 + 2^2 + 5^3 + 3^4 + 1^5 = 215$.

Петата заявка е от тип 3, т.е. вземи обратната. След тази заявка пермутацията се превръща в $[5, 2, 4, 1, 3]$. *Стойността* на тази пермутация е $5^1 + 2^2 + 4^3 + 1^4 + 3^5 = 317$.

Последната заявка е от тип 1, т.е. разшири. След тази заявка пермутацията се превръща в $[3, 1, 4, 2, 5]$. *Стойността* на тази пермутация е $3^1 + 1^2 + 4^3 + 2^4 + 5^5 = 3209$.

Scoring

Тестовете към тази задача се състоят от 12 групи. Точките за всяка група се дават само ако са преминати всички тестове от групата и всички тестове от някои от предходните групи. Обърнете внимание, преминаването на тестовете от условието не е необходимо за някои от групите. Общия брой точки за всяка група е равен на максималния брой точки, получени за тази група тестове от всички събмити.

Група	Точки	Доп. ограничения		Необх. групи	Коментари
		n	q		
0	0	–	–	–	тестовете от условието.
1	15	$n \le 1000$	$q \le 1000$	0	
2	22	–	–	–	$b_i = b_j$ за всички $1 \le i, j \le q$
3	26	–	–	–	$b_i \le 2$ за всички $1 \le i \le q$
4	16	–	–	–	$p_i = i$ за всички $1 \le i \le n$
5	21	–	–	0 – 4	

Problem Toyota. Задачи от Сашко

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	3 seconds
Memory limit:	512 megabytes

Сашко наскоро се премести в многоетажна сграда. В сградата има n етажа, номерирани с числата от 1 до n. На всеки етаж от къщата живее точно един жител. Между етажите са построени $n-1$ стълби, при това не е задължително те да са построени между съседни етажи. Известно е, че за всеки етаж, освен първия, има точно една стълба, водеща към по-нисък етаж. За i-тия етаж ($2 \le i \le n$) такава стълба води до етаж с номер p_i.

Сашко се кани да реши k задачи, номерирани от 1 до k. За задача с номер i Сашко е изчислил, ченай-оптимално е да я решава на етаж с номер x_i. Тъй като задачите се различават помежду си всички x_i са различни.

Да решаваш задачи сам е скучно, затова за всяка задача Сашко иска да покани поне един жител, за да я решат заедно. Обаче, жителите на сградата никак не обичат катеренето по стълби и са склонни само да слизат до етажите, където ще бъдат решавани задачите. Затова Сашко може да покани жителя от етаж j да решава задачата i само ако от j-тия етаж има възможност да се достигне до етаж x_i, използвайки няколко, възможно нула, стълби, водещи всеки път до етаж с по-малък номер. По този начин жителя от етажа j може да решава задачата i само ако $j = x_i$ или $p_j = x_i$, или $p_{p_j} = x_i$ и т.н.

Жителите никак не обичат да слизат по стълбите без нужда. Затова, ако Сашко покани някаква група хора да решават задача, те са готови да се съберат да решават задачата **на най-високия етаж, до който могат да се спуснат**. Например, ако от етаж 3 води стълба към етаж 2, Сашко не може да покани жителите на етажи 2 и 3 да решават задача на етаж 1, тъй като всички жители могат да се съберат на по-горен етаж.

Саша не обича да изглежда нахален и затова ще помоли жителите от всеки етаж да решат само по една задача. На някои етажи обаче Сашко може да не иска от живущите да решават задачи.

Сашко има друга любима задача, която няма да сподели с никого освен с вас. Но за да ви разкаже за нея, трябва да му помогнете да изчисли броя на различните начини да покани жителите да решават задачи с него, така че да бъдат изпълнени всички ограничения. Два начина се считат за различни, ако поне една задача е решена от различни групи жители.

Input

Първият ред на стандартния вход съдържа две цели числа n и k ($3 \le n \le 10^6$, $1 \le k \le \min(n, 2000)$) — броя етажи в сградата и броя задачи.

Вторият ред съдържа k цели числа x_1, x_2, …, x_k ($1 \le x_i \le n$) — етажите, на които Сашко ще решава задачи. Гарантирано е, че всички x_i са различни.

Третия ред съдържа $n-1$ цели числа p_2, p_3, …, p_n ($1 \le p_i < i$), където p_i описва номера на етажа, към който води стълба, която слиза от етаж i.

Output

Изведете едно число — броя различни начини да се поканят жителите на сградата да решават задачи заедно със Сашко по модул 998 244 353.

Examples

input	output
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

Note

В първия пример за Сашко има пет начина да покани жителите на сградата да решават задачи:

- само с жителя на етаж 1;
- с жителите на етажи 1 и 2;
- с жителите на етажи 1 и 3;
- с жителите на етажи 1, 2 и 3;
- с жителите на етажи 2 и 3;

Сашко не може да покани да решава задачи само жителя от етаж 2 тъй като тогава най-високия етаж, на който могат да се съберат всички жители, желаещи да решават задачи, ще бъде етаж 2, а Сашко иска да решава задачи на етаж 1.

Във втория пример, два различни подходящи начина да се поканят жителите да решават задачи биха могли да бъдат следните:

- Да се поканят жителите на етажи 2 и 6 да решават първата задача, а жителя от етаж 5 да решава втората задача.
- да се покани жителя от етаж 2 да решава първата задача, а жителите от етажи 5 и 6 да решават втората задача.

Scoring

Тестовете към тази задача се състоят от 9 групи. Точките за всяка група се дават само ако са преминати всички тестове от групата и всички тестове от някои от предходните групи. Обърнете внимание, преминаването на тестовете от условието не е необходимо за някои от групите. **Offline-проверка** означава, че резултатите от тестването на вашите решения за дадена група ще бъдат достъпни след края на състезанието. Общия брой точки за всяка група е равен на максималния брой точки, получени за тази група тестове от всички събмити.

Група	Точки	Допълнителни ограничения		Необх. групи	Коментари
		n	k		
0	0	–	–	–	Тестовете от условията
1	12	$n \leq 10$	$k \leq 10$	0	
2	13	$n \leq 500$	$k \leq 500$	0, 1	
3	9	–	$k = 1$	–	
4	10	–	–	–	$p_i = i - 1$
5	13	–	–	4	Всеки етаж е свързан с не повече от два етажа с с по-голям номер
6	14	$n \leq 200\,000$	$k \leq 500$	0 – 2	
7	11	–	$k \leq 500$	0 – 3, 6	
8	10	–	$k \leq 1000$	0 – 3, 6, 7	
9	8	–	–	0 – 8	**Offline-проверка**

Problem Mercedes. Вълнение преди олимпиадата

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

This problem was translated using automatic translator. If some details are unclear, refer to english statements.

Преди входа на затворената олимпиада се събраха n участници, номерирани от 1 до n. Известно е, че всяка минута в залата за състезания ще бъде пуснат по един участник в реда на нарастващите им номера. След една минута в залата за състезания ще влезе първият участник, след две минути — вторият участник и така нататък. По този начин, i-ят участник ще влезе в залата за състезания след i минути след началото на пускането на участниците в залата за състезания.

Всеки участник има определено ниво на вълнение преди олимпиадата. Нивото на вълнение на всеки участник е зададено с някакво цяло (възможно отрицателно) число. Преди началото на пускането на участниците, нивото на вълнение на участника с номер i е равно на a_i. Всяка минута нивото на вълнение на участника ще се променя с величина b_i. По този начин, след x минути след началото на пускането на участниците, нивото на вълнение на участника с номер i ще бъде равно на $a_i + x \cdot b_i$.

Александър — опитен психолог, който реши да успокои участниците в опашката за вход на олимпиадата. За целта Александър разговаря с участниците и ги успокоява. С всеки участник Александър може да разговаря не повече от един път. След общуването с Александър нивото на вълнение на участника става равно на 0 и спира да се променя след това. *Ефективността* на работата на Александър с участника с номер i се счита за величината на вълнението на участника в момента на общуването с Александър. По този начин, ако Александър разговаря с участника i след t_i минути след началото на пускането на участниците на олимпиадата, *ефективността* ще бъде равна на $a_i + t_i \cdot b_i$. Обърнете внимание, че ако нивото на вълнение на участника е било отрицателно, то и *ефективността* на работата ще бъде отрицателна.

Александър ще работи с участниците в реда на нарастващите им номера. Въпреки това, Александър не е задължен да говори с всички участници, т.е. е възможно Александър да не разговаря с последните участници в опашката. Обърнете внимание, че с всеки участник Александър ще разговаря не повече от един път и това трябва да се случи преди момента на входа на участника в залата за състезания. В същото време, всяка минута Александър може да разговаря веднага с няколко участници. По-формално, процесът на работа на Александър може да бъде описан по следния начин:

- Александър ще разговаря с първите k участници в опашката, където k той избира сам.
- За всеки от първите k участници фиксираме цяло неотрицателно число t_i — времето на общуването с Александър. Обърнете внимание, че t_i може да бъде равно на нула, което означава, че Александър е говорил с i-я участник преди да пуснат първия участник в залата за състезания.
- За всеки i от 1 до k, $t_i < i$, тъй като Александър трябва да разговаря с участниците преди техния вход в залата за състезания.
- За всеки i от 1 до $k-1$, $t_i \le t_{i+1}$, тъй като Александър разговаря с участниците в реда на нарастващите им номера.
- *Общата ефективност* на общуването на Александър с участниците ще бъде описана от следната формула:

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

За Александър предварително е фиксиран план за работа с участниците. Планът за работа с участниците е зададен с последователност от n цели числа $p_1, p_2, \ldots, p_n$. За всеки i, ако $p_i \neq -1$, то в

момента на пускането на първите i участници в залата за състезания (след i минути след началото на пускането на участниците), Александър трябва да говори с първите p_i участници и с никого повече. В този случай също така е гарантирано, че $p_i \geq i$. Ако $p_i = -1$, то това означава, че няма ограничения за броя участници, с които Александър трябва да работи след i минути след началото на пускането на участниците.

По-формално, ако $p_i \neq -1$, то:

- $p_i \geq i$;
- $t_{p_i} < i$;
- За всяко j, такова че $p_i < j \leq k$ е вярно, че $t_j \geq i$.

Помогнете на Александър да определи каква може да бъде максималната величина на *общата ефективност* на работата с участниците при изпълнение на всички ограничения. Гарантира се, че отговорът винаги съществува.

Input

Първият ред съдържа единствено цяло число n ($1 \leq n \leq 10^6$) — броя на участниците, които стоят в опашка за вход в залата за състезания на олимпиадата.

Следващите n реда съдържат по две цели числа a_i и b_i ($-10^9 \leq a_i \leq 10^9$, $-10^6 \leq b_i \leq 10^6$) — параметри на вълнението на i-я участник.

Следващият ред съдържа n цели числа p_1, p_2, …, p_n ($i \leq p_i \leq n$ или $p_i = -1$) — описание на плана за работа на Александър.

Гарантира се, че за всяка двойка $1 \leq i < j \leq n$ е изпълнено $p_i \leq p_j$, ако $p_i \neq -1$ и $p_j \neq -1$.

Output

Изведете едно число — максималната възможна *обща ефективност* на работата на Александър.

Може да се покаже, че Александър винаги ще може да изпълни работата си, спазвайки всички допълнителни ограничения и план за работа.

Examples

input	output
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

Note

В първия тестов пример е оптимално да се избере $k = 4$ и $t = \{0, 0, 0, 3\}$. Тогава *общата ефективност* ще бъде равна:

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

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

Във втория тестов пример е оптимално да се избере $k = 3$ и $t = \{0, 0, 1\}$. Тогава *общата ефективност* ще бъде 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 третия тестов пример е оптимално да се избере $k = 3$ и $t = \{0, 1, 2\}$. Тогава *общата ефективност* ще бъде равна:

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

Scoring

Тестовете по тази задача се състоят от 14 групи. Точките за всяка група се присъждат само при преминаване на всички тестове на групата и всички тестове на някои от предишните групи. Обърнете внимание, че преминаването на тестовете от условието не е задължително за някои групи. **Offline проверка** означава, че резултатите от тестването на вашето решение по тази група ще станат достъпни само след края на състезанието. Крайната точка за всяка група е равна на максималната точка, получена за тази група тестове от всички изпратени решения.

Група	Точки	Ограничения			Необходими групи	Коментар
		n	b_i	p_i		
0	0	–	–	–	–	Тестове от условието.
1	6	$n \leq 100$	–	$p_i = -1$	–	
2	6			–	0 – 1	
3	7	$n \leq 5000$	–	$p_i = -1$	1	
4	6			–	0 – 3	
5	7	–	$b_i \leq 0$	$p_i = -1$	–	
6	5			–	5	
7	7	–	$b_i \geq 0$	$p_i = -1$	–	
8	5			–	7	
9	9	–	$b_i \leq b_{i+1}$	$p_i = -1$	–	
10	8			–	9	
11	10	$n \leq 100\,000$	–	$p_i = -1$	–	Брой $b_i > 0$ не повече от 10
12	7			–	11	
13	9	–	–	$p_i = -1$	1, 3, 5, 7, 9, 11	**Offline проверка**
14	8			–	0 – 13	