

Загальна інформація

Загальне

- Вам дано 5 годин на рішення 5 задач, кожна з яких оцінюється в 100 балів.
- Ви можете зробити до 60 відправлень. Немає значення, як саме ви використаєте ці відправлення по задачах. Ви можете відправити будь-яку кількість рішень на кожную задачу, але сумарна кількість відправлень по всім задачам не має перевищувати 60.
- Журі не гарантує, що існують розв'язки на повний бал на таких мовах, як Pascal, Python та Java.
- Під час олімпіади суворо забороняється використовувати інтернет, за виключенням сайту, на якому ви працюєте. Не можна використовувати будь-які переносні носії інформації.
- Результати олімпіади будуть доступні на сайті `oi.in.ua` після змагання.

Оцінювання

Є два види оцінювання:

- «Потестове оцінювання». Кожний тест оцінюється незалежно від інших. Проходження тесту приносить певну кількість балів. Приклади оцінюються в 0 балів.
- «Блочне оцінювання». Усі тести поділені на блоки, які описані в умові задачі. Бали нараховуються лише при проходженні **всіх** тестів блоку. Якщо обмеження блока i не менші за обмеження блока j , то для нарахування балів за блок i , також потрібно, щоб пройшли всі тести блока j . В умові про це не буде сказано. Також є «нульовий блок», який складається з прикладів, він оцінюється в 0 балів. В умові про це згадувати не будуть.

Види задач

Є два види задач:

- «З вводом та виводом даних». Вам потрібно зчитати дані з файлу типу «`problem.in`», рішення задачу та вивести результат у файл «`problem.out`». Зверніть увагу, що ім'я файлів змінюється в залежності від задачі.
- «З модулями». Вам потрібно реалізувати функції, які описані у задачі. Зверніть увагу, що в цьому виді задач **суворо забороняється** зчитувати та виводити дані. Ви маєте працювати з даними лише в той спосіб, який описаний в умові задачі. Вам дадуть архів, в якому буде три файли для кожної мови програмування: `header`, `sample`, `footer`. Вам потрібно змінити `sample` та відправити лише його в систему. Під час компіляції вашого рішення, код в `header` вставляється перед вашим кодом, а код в `footer` після. Тобто формується новий файл, який починається з `header`, потім йде ваше рішення, і лише після нього `footer`. Зверніть увагу, що `footer`, який використовується при тестуванні, може відрізнитись від того, який буде в архіві.

Питання

- Ви можете ставити питання за допомогою системи. Зверніть увагу, що питання мають бути такими, на які можна відповісти «Так» або «Ні». Ви можете отримати одну з наступних відповідей:
 - «Відповідь в умові», якщо на ваше питання можна відповісти, прочитавши умову або загальну інформацію про тур.
 - «Без коментарів», якщо питання стосується інформації, яку журі не бажає розголошувати, наприклад, методу розв'язання задачі.
 - «Незрозуміле питання», якщо журі не зрозуміло ваше питання.
 - «Так» або «Ні».

Задача А. Газон у Потоколяндії

Назва вхідного файлу:	lawn.in
Назва вихідного файлу:	lawn.out
Обмеження використання часу:	0.25 seconds
Обмеження використання пам'яті:	256 megabytes

У саду Потоколяндії є газон шириною в 1 метр та довжиною в n метрів.

Козак Вус хоче покосити цей газон за допомогою наступних дій: він вибирає деякий підвідрізок цього газону довжиною k (частина газону шириною в 1 метр та довжиною в k метрів) такий, що він повністю покритий травою та косить траву цієї частини газону. На цю дію йому необхідно витратити одну годину.

Також відомо, що рівно через годину з моменту, коли було завершено описану дію, трава на цій ділянці знову виросте. Це виростання відбувається миттєво в описаний момент часу.

Тоді Козак Вус зрозумів, що його задовольнить навіть не повністю покошений газон — достатньо щоб будь-яка ділянка газону була покосена хоча б один раз.

Допоможіть Козаку дізнатись мінімальний час, який йому знадобиться, щоб будь-яка ділянка газону була покосена хоча б один раз.

Початково на всьому газоні трава не є покосеною.

Формат вхідних даних

Перший рядок містить два цілих числа n та k ($1 \leq k \leq n \leq 100$) — довжина газону в метрах та довжина підвідрізка, який вибирається для того, щоб покосити газон, відповідно.

Формат вихідних даних

Виведіть одне ціле число — відповідь на задачу у годинах. Легко показати що відповідь завжди є цілим числом.

Приклади

lawn.in	lawn.out
5 2	3
3 3	1

Примітка

Пояснення до першого прикладу:



Пояснення до другого прикладу:

Козак Вус може покосити весь газон за одну годину.

Оцінювання

Кожний тест, крім прикладів, оцінюється в 4 бали.

Задача В. Мафія у Потоколяндії

Назва вхідного файлу:	mafia.in
Назва вихідного файлу:	mafia.out
Обмеження використання часу:	0.25 seconds
Обмеження використання пам'яті:	256 megabytes

Нещодавно у Потоколяндії розробили онлайн версію гри «Мафія».

Оскільки Козак Вус — головний програміст Потоколяндії, то саме йому і довірили протестувати цю гру.

Всім відомо, що головне у грі «Мафія» — чати повідомлень. Козак Вус вважає гру протестованою, якщо у чатах сумарно буде написано хоча б n слів.

Оскільки програмісти не тестують чати, Козаку доведеться найняти кілька тестувальників на наступних умовах: кожен тестувальник напише привітальне повідомлення, що складатиметься з a слів у загальний чат гри, а також напише кожному іншому тестувальнику привітальне повідомлення, що складатиметься з b слів у особистий чат. Тестувальники не писатимуть інших повідомлень, крім описаних вище.

Припустимо, що є 3 тестувальники, кожен з яких має написати «Усім привіт!» (2 слова) у загальний чат, а також «Привіт!» (1 слово) кожному іншому тестувальнику в особистий чат. Тоді у загальному чаті буде написано $2 \cdot 3 = 6$ слів (кожен тестувальник напише 2 слова), а в кожному особистому чаті буде написано по два слова (по одному слову від кожного тестувальника). Оскільки всього 3 особистих чатів (між першим та другим, між першим та третім, між другим та третім тестувальниками), то всього буде написано $3 \cdot 2 = 6$ слів в особистих чатах. Отже, всього буде написано $6 + 6 = 12$ слів у всіх чатах.

Під час тестування лише тестувальники писатимуть повідомлення у чати повідомлень.

Через те, що Козак Вус не хоче витратити багато грошей на найм тестувальників, він вирішив мінімізувати їх кількість. Допоможіть йому дізнатись мінімальну кількість тестувальників, яку доведеться найняти, щоб у чатах сумарно було написано хоча б n слів.

Формат вхідних даних

Перший рядок містить три цілих числа n , a та b ($1 \leq n \leq 10^{12}$, $0 \leq a, b \leq 10^6$, $0 < a + b$) — мінімальна кількість слів, яка має бути написана, кількість слів, яку має написати кожний тестувальник у загальний чат, а також кількість слів, яку має написати кожний тестувальник кожному іншому в особистий чат.

Формат вихідних даних

Виведіть одне ціле число — мінімальну кількість тестувальників, яку доведеться найняти, щоб у чатах сумарно було написано хоча б n слів.

Приклади

mafia.in	mafia.out
10 1 3	3
10 8 2	2

Примітка

У першому прикладі, якщо буде найнято 3 тестувальника, то у загальний чат буде написано 3 слова, а в особистих чатах буде написано 18 слів.

У другому прикладі, якщо буде найнято 2 тестувальника, то у загальний чат буде написано 16 слів, а в особистих чатах буде написано 4 слова.

Оцінювання

Кожний тест, крім прикладів, оцінюється в 5 балів.

Задача С. Козак Вус та мінйони

Назва вхідного файлу:	<code>minions.in</code>
Назва вихідного файлу:	<code>minions.out</code>
Обмеження використання часу:	0.75 seconds
Обмеження використання пам'яті:	256 megabytes

Бути мінйоном — це звісно ж круто, але Козак Вус сильніший за них...

У розпорядженні у Козака Вуса є n мінйонів, пронумерованих цілими числами від 1 до n . Кожен мінйон характеризується своїми силою та витривалістю. У i -го мінйона сила рівна a_i , а витривалість рівна b_i .

Козак Вух попросив Козака Вуса передати йому в розпорядження загін мінйонів. Припустимо, що Козак Вух попросив передати k мінйонів. Тоді Козак Вус може передати **будь-які** k мінйонів своєму другу. Ці мінйони можуть мати будь-які номери, не обов'язково послідовні. В кожному загоні має бути один лідер, який вибирається Козаком Вусом.

Герої для кожного загону мінйонів визначили його ціну. Вони вважають, що ціна загону рівна сумі сили лідера та витривалостей всіх **інших** мінйонів в загоні (якщо такі є). Розміром загону називають кількість мінйонів у ньому.

Припустимо, що є чотири мінйони, у яких:

1. сила 3, а витривалість 6,
2. сила 7, а витривалість 3,
3. сила 1, а витривалість 8,
4. сила 6, а витривалість 5.

Якщо Козак Вух просить передати йому загін з 3 мінйонів, то Козак Вус може, наприклад, вибрати першого, другого та четвертого. Якщо він назначить другого лідером, то ціна загону буде рівна 7 (сила лідера) $+ 6$ (витривалість першого) $+ 5$ (витривалість четвертого) $= 18$.

Нехай $f(k)$ — **мінімальна** ціна загону серед всіх загонів, в яких розмір k .

У попередньому прикладі Козаку Вусу для мінімізації ціни краще за все вибрати загін, в якому другий, третій та четвертий мінйони та призначити третього мінйона лідером. Тоді ціна загону рівна $1 + 3 + 5 = 9$. Оскільки немає кращого рішення, то $f(3) = 9$.

Тепер Козаки хочуть дізнатись всі можливі значення f . Допоможіть Козакам дізнатись значення $f(1), f(2), \dots, f(n)$.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 2 \cdot 10^5$) — кількість мінйонів у розпорядженні у Козака Вуса.

Кожний з наступних n рядків містить два цілих числа a_i та b_i ($1 \leq a_i, b_i \leq 10^9$) — значення сили та витривалості мінйона з номером i відповідно.

Формат вихідних даних

Виведіть n цілих чисел — $f(1), f(2), \dots, f(n)$.

Приклади

minions.in	minions.out
3 1 4 3 3 2 1	1 2 5
5 3 4 3 3 2 4 4 3 5 3	2 5 8 11 15

Примітка

Пояснення до першого прикладу:

Щоб утворити загін мінімальної вартості серед всіх загонів розміру 1, необхідно утворити загін що складається з міньюна з номером 1.

Щоб утворити загін мінімальної вартості серед всіх загонів розміру 2, необхідно утворити загін що складається з міньюнів з номерами 1 та 3 та вибрати лідером міньюна з номером 1.

Щоб утворити загін мінімальної вартості серед всіх загонів розміру 3, необхідно утворити загін що складається з всіх міньюнів та вибрати лідером міньюна з номером 1.

Оцінювання

№	Обмеження			Бали
	n	a, b	Додаткові	
1	$1 \leq n \leq 2 \cdot 10^5$	$1 \leq a, b \leq 3$	-	7
2	$1 \leq n \leq 2000$	$1 \leq a, b \leq 10^9$	Всі значення a рівні між собою	13
3	$1 \leq n \leq 2 \cdot 10^5$			8
4	$1 \leq n \leq 100$			23
5	$1 \leq n \leq 2000$			19
6	$1 \leq n \leq 2 \cdot 10^5$			$1 \leq a, b \leq 2 \cdot 10^5$
7	$1 \leq n \leq 2 \cdot 10^5$	$1 \leq a, b \leq 10^9$		13

Задача D. Футбол у Потоколяндії

Обмеження використання часу: 0.75 seconds
 Обмеження використання пам'яті: 256 megabytes

Збірна команда Потоколяндії з футболу складається з n футболістів, пронумерованих цілими числами від 0 до $n - 1$. Для кожного гравця відоме його вміння грати у футбол (у Потоколяндії ця характеристика завжди є цілим числом). Вміння гравця з номером i рівне a_i .

Головний тренер Козак Вус часто проводить різні тренування. Під час тренування вміння гравців можуть змінюватись. Але оскільки Козак не дуже досвідчений тренер, він завжди тренує лише пару футболістів, чия різниця номерів рівна 1. При цьому обов'язково, щоб сума вмінь гравців які тренуватимуться після тренування, залишалась такою самою, як і до тренування. Наприклад, якщо тренуватимуться гравці з номерами x та $x + 1$ та до вміння гравця з номером x буде додано деяке число k то від вміння гравця з номером $x + 1$ буде віднято k .

Згідно з новим регламентом заміни під час матчу заборонені та футболісти, які братимуть участь у матчі, мають мати послідовні номери. Іншими словами, номери футболістів що братимуть участь у матчі, можна буде описати парою цілих чисел (l, r) , що означатиме, що братимуть участь у матчі лише футболісти з номерами $l, l + 1, \dots, r$.

Козак Вус вважає, що перемогти збірну Сішляндії у фіналі чемпіонату України можна лише у випадку, якщо сума вмінь гравців що братимуть участь у матчі буде рівною w .

У певному матчі чемпіонату лише гравці з номерами $a_i, a_i + 1, \dots, b_i$ згодні грати в матчі.

Тепер Козак просить Вас допомогти йому знаходити команду, яка має шанси на перемогу у матчі проти збірної Сішляндії, якщо відбуваються тренування та не всі гравці згодні грати в певному матчі.

Протокол взаємодії

Вам потрібно реалізувати три функції (тут використовується **псевдокод**, щоб дізнатись деталі реалізації для вашої мови, дивіться нижче):

```
void init(integer n, array of integers a, integer w, integer group)
```

- n — кількість футболістів;
- a — масив вмінь гравців;
- w — сума вмінь гравців, які братимуть участь у матчі, яка задовольнить Козака Вуса;
- $group$ — номер блока;
- ця функція викликається першою лише один раз. Лише після виклику цієї функції, будуть викликатись інші дві.

```
void upd(integer p, integer k)
```

- p — мінімальний з номерів гравців, що братимуть участь у тренуванні;
- k — значення, що додається до вміння гравця з номером p та відніметься від вміння гравця з номером $p + 1$;
- ця функція викликається, коли Козак Вус проводить тренування.

```
longinteger get(integer a, integer b)
```

- a_i, b_i — параметри, що означають, що в даний момент грати проти збірної Сішляндії погоджуються лише футболісти з номерами $a_i, a_i + 1, \dots, b_i$;
- ця функція викликається, коли потрібно зібрати команду футболістів, яка має шанси на перемогу у матчі проти збірної Сішляндії;

- функція має повернути ціле число $(l \cdot 10^5 + r)$, де (l, r) — пара цілих чисел, для якої виконується $0 \leq l \leq r < n$, що означає, що команда що складатиметься з гравців з номерами $l, l+1, \dots, r$ має шанси на перемогу у матчі проти збірної Сішляндії; якщо ж неможливо зібрати таку команду, ця функція має повернути число -1 . Якщо правильних відповідей декілька, дозволяється повернути будь-яку з них. Зауважте, що команда Потоколяндії має містити хоча б одного футболіста.

Зверніть увагу, що футболісти нумеруються починаючи з 0, а не з 1.

В залежності від того, на якому сервері ви пишете, ви можете завантажити архів за одним з наступних посилань:

- <http://kremped.org.ua/files/uoi-2019-region-tour-2-lkwdkqnfsw.zip>
- <http://ejudge.sumdu.edu.ua/statements/uoi-2019-region-tour-2-sqowddas.zip>
- <http://olymp.uzhnu.edu.ua/statements/uoi-2019-region-tour-2-fjqweqrtpzd.zip>
- <http://dn.hoippo.km.ua/statements/tur2/34a6f833c98c21079bbccc3adee00a28.zip>

Формат вхідних даних

Перший рядок містить чотири цілих числа $n, m, w, group$ ($1 \leq n, m \leq 10^5, |w| \leq 10^9$) — кількість членів збірної команди Потоколяндії з футболу, кількість подій (сума кількостей тренувань та запитів знаходження команди, що має шанси на перемогу), значення w та номер блока відповідно.

Другий рядок містить n цілих чисел a_0, a_1, \dots, a_{n-1} ($|a_i| \leq 10^9$) — значення вмінь футболістів.

Кожний з наступних m рядків описує подію і має один з наступних форматів:

1. «1 p k» ($0 \leq p < n - 1, |k| \leq 2 \cdot 10^9$) — мінімальний з номерів гравців, що братимуть участь у тренуванні, та значення, що додається до вміння гравця з номером p та відніметься від вміння гравця з номером $p + 1$.
2. «2 a b» ($0 \leq a \leq b < n$) — параметри, що означають, що в даний момент грати проти збірної Сішляндії погоджуються футболісти з номерами $a_i, a_i + 1, \dots, b_i$.

Гарантується, що в будь-який момент часу для усіх i виконується $|a_i| \leq 10^9$ та що буде принаймні одна подія другого типу.

Формат вихідних даних

Для кожної події другого типу буде виведено пару чисел (l, r) , яка визначатиме гравців команди, яка має шанси на перемогу проти збірної Сішляндії, або -1 , якщо таку команду неможливо утворити за відповідних умов.

Приклад

input	output
14 9 3 0	0 1
1 2 -1 5 0 3 5 -2 -2 4 1 4 6 0	8 10
2 0 1	-1
2 0 13	9 10
1 5 7	0 1
2 4 8	-1
1 10 -2	
2 6 11	
2 0 5	
1 1 -1	
2 0 9	

Примітка

Пояснення до прикладу:

Значення елементів масиву a :

Після 1-го тренування — [1, 2, -1, 5, 0, 10, -2, -2, -2, 4, 1, 4, 6, 0];

Після 2-го тренування — [1, 2, -1, 5, 0, 10, -2, -2, -2, 4, -1, 6, 6, 0];

Після 3-го тренування — [1, 1, 0, 5, 0, 10, -2, -2, -2, 4, -1, 6, 6, 0].

Оцінювання

№	Обмеження		Бали
	n, m	Додаткові	
1	$1 \leq n, m \leq 100$	-	12
2	$1 \leq n, m \leq 5000$		13
3	$1 \leq n, m \leq 10^5$	В будь-який момент часу всі значення a_i невід'ємні Всього відбудеться не більше 500 подій другого типу	11
4		Не проводиться жодних тренувань	15
5		В будь-який момент часу всі значення a_i рівні 0 або 1	12
6		$w=0$	13
7		-	24

Деталі реалізації для C

Реалізуйте три функції:

```
void init(int n, int *a, int w, int group)
```

```
void upd(int p, int k)
```

```
long long get(int a, int b)
```

Деталі реалізації для C++

Реалізуйте три функції:

```
void init(int n, vector<int> a, int w, int group)
```

```
void upd(int p, int k)
```

```
long long get(int a, int b)
```

Деталі реалізації для Java

Вам потрібно реалізувати class `Test`, в якому мають бути три функції:

```
public static void init(int n, List<Integer> a, int w, int group)
```

```
public static void upd(int p, int k)
```

```
public static long get(int a, int b)
```

Деталі реалізації для Python

Реалізуйте три функції:

```
def init(n, a, w, group):
```

```
def upd(p, k):
```

```
def get(a, b):
```

Деталі реалізації для Pascal

Реалізуйте три функції:

```
procedure init(n: longint; a: array of longint; w: longint; group: longint);
```

```
procedure upd(p, k: longint);
```

```
function get(a, b: longint): int64;
```


Задача Е. Козак Вус та картопля

Назва вхідного файлу:	potato.in
Назва вихідного файлу:	potato.out
Обмеження використання часу:	4 seconds
Обмеження використання пам'яті:	256 megabytes

Як відомо, Потоколяндія складається з n міст, що з'єднані m односторонніми дорогами. Міста пронумеровані від 1 до n . У місті з номером i знаходиться p_i мішків з картоплею. Для кожної дороги відоме деяке ціле значення w_i — час, необхідний для того, щоб Козак переніс один мішок картоплі цією дорогою. Для різних доріг значення w можуть відрізнятися. Можна вважати, що у кожному з міст неймовірно велика кількість Козаків та будь-яку кількість мішків з картоплею можна переносити одночасно.

Козак Вус дізнався що скоро у Потоколяндію потрапить вірус, який знищує картоплю. Щоб захистити врожай Козаки збираються заховати картоплю у бункери.

Всього у країні s бункерів, пронумерованих від 1 до s . Бункер з номером i розташований у місті з номером t_i та може зберегти від вірусу до c_i мішків з картоплею.

Тепер герою стало цікаво дізнатись мінімальний час, за який Козаки зможуть заховати всі мішки з картоплею у бункери. Допоможіть йому з цим завданням!

Формат вхідних даних

Перший рядок містить три цілих числа n , m та s ($1 \leq n \leq 10^5$, $0 \leq m \leq 6 \cdot 10^5$, $1 \leq s \leq 18$) — кількість міст, доріг та бункерів у Потоколяндії відповідно.

Другий рядок містить n цілих чисел p_1, p_2, \dots, p_n ($0 \leq p_i \leq 10^9$) — кількість мішків з картоплею у містах.

Кожний з наступних m рядків описує дорогу трьома цілими числами u_i, v_i, w_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq w_i \leq 10^9$) — номери міст, які з'єднує відповідна дорога та час необхідний для того, щоб Козак переніс один мішок картоплі цією дорогою. Дорога з номером i дає змогу пересуватись від міста з номером u_i до міста з номером v_i . Гарантується, що для будь-яких u_i, v_i існує не більше однієї дороги що веде від міста u_i до міста v_i .

Кожний з наступних s рядків описує бункер двома цілими числами t_i та c_i ($1 \leq t_i \leq n$, $1 \leq c_i \leq 10^9$) — номер міста, де розташовано відповідний бункер та максимальна кількість мішків з картоплею, які може вберегти від вірусу цей бункер.

Формат вихідних даних

Виведіть одне ціле число — мінімальний час, за який Козаки зможуть заховати всі мішки з картоплею у бункери. Якщо неможливо заховати всі мішки з картоплею у бункери виведіть -1 .

Приклади

potato.in	potato.out
2 1 1 3 2 2 1 4 1 6	4
4 6 2 2 0 0 2 2 1 6 3 1 2 3 2 3 1 3 4 4 3 4 2 4 6 3 2 2 2	7
7 10 3 0 1 1 1 1 0 2 2 1 1 3 2 1 3 1 1 6 4 5 4 5 9 3 4 1 7 6 10 5 7 3 6 5 3 4 3 1 6 5 1 1 2 1	22

Примітка

Пояснення до першого прикладу:

У бункер можна заховати всі мішки з картоплею. Для цього достатньо перенести з міста з номером 2 до міста з номером 1 два мішки з картоплею.

Пояснення до другого прикладу:

Доведеться заховати по 2 мішки з картоплею у кожен з бункерів. Для цього достатньо перенести з міста з номером 1 до міста з номером 3 два мішки з картоплею та перенести з міста з номером 4 до міста з номером 2 два мішки з картоплею (це перенесення краще здійснити по такому маршруту: $4 \rightarrow 3 \rightarrow 2$).

Оцінювання

№	Обмеження			Бали		
	n, m	s	Додаткові			
1	$m = 2 \cdot (n - 1)$ $1 \leq n \leq 1000$	$s = 1$	Для будь-якого $1 \leq i < n$ існує дорога між містами з номерами i та $i + 1$	$t_1 = 1$	Для будь-якої дороги, що веде з міста u до	4
2			-	з міста v , існує дорога, що веде	4	
3	З будь-якого міста можна дістатись до будь-якого іншого		з міста v до міста u , і у цих доріг рівні значення w	7		
4	$1 \leq n \leq 1000$ $0 \leq m \leq 4000$		-	12		
5	$1 \leq n \leq 10^5$		Всі значення w рівні 1	7		
6	$0 \leq m \leq 4 \cdot 10^5$		-	10		
7			$1 \leq s \leq 3$	10		
8	$1 \leq n \leq 1000$ $0 \leq m \leq 4000$	$1 \leq s \leq 10$	-	9		
9	$1 \leq n \leq 10^5$			9		
10	$0 \leq m \leq 6 \cdot 10^5$	$1 \leq s \leq 14$		14		
11		$1 \leq s \leq 18$		14		