

Загальна інформація

Загальне

- На офіційному турі, вам буде дано 5 годин на рішення 5 задач, кожна з яких оцінюється в 100 балів.
- На тренувальному турі вам дані задачі, щоб ви ознайомились з системою. Результати тренувального туру **ніяк** не впливають на ваш результат на олімпіаді. Не зважаючи на це, ми **наполегливо рекомендуємо** вам рішення **всі** задачі.
- Ви можете зробити 60 відправлень. Немає значення, як саме ви використаєте ці відправлення по задачах. Ви можете відправити будь-яку кількість рішень на кожну задачу, але сумарна кількість відправлень по всім задачам не має перевищувати 60.

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».

Зворотний зв'язок

Ви можете надсилати свої питання про тренувальний тур або в системі, або на пошту ukr.oi.help@gmail.com.

Задача А. Сума

Назва вхідного файлу: `aplusb.in`
Назва вихідного файлу: `aplusb.out`
Обмеження використання часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Потрібно знайти суму чисел a та b .

Формат вхідних даних

Перший рядок містить два цілих числа a та b ($-10^{18} \leq a, b \leq 10^{18}$) — два цілих числа, які потрібно додати.

Формат вихідних даних

Виведіть суму двох чисел.

Приклади

<code>aplusb.in</code>	<code>aplusb.out</code>
10 6	16
-56 42	-14

Оцінювання

Кожний тест, крім вхідних, оцінюється в 5 балів.

Задача В. Сума масиву

Назва вхідного файлу: `array-sum.in`
Назва вихідного файлу: `array-sum.out`
Обмеження використання часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Дано n цілих чисел a_1, a_2, \dots, a_n . Знайдіть їх суму.

Формат вхідних даних

Перший рядок містить ціле число n ($1 \leq n \leq 2 \cdot 10^6$) — кількість чисел.

Другий рядок містить n цілих чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — числа масиву.

Формат вихідних даних

Виведіть одне число — суму масиву.

Приклад

<code>array-sum.in</code>	<code>array-sum.out</code>
5 4 1 8 1 1	15

Підзадача 1

Оцінюється у 10 балів. Також виконуються обмеження:

- $n \leq 1\,000$;
- $a_i \leq 1\,000$.

Підзадача 2

Оцінюється у 10 балів. Також виконується обмеження:

- $n \leq 10\,000$.

Підзадачі 3-12

Кожна підзадача оцінюється у 8 балів. У третій підзадачі $n \leq 200\,000$, у четвертій $n \leq 400\,000$, у п'ятій $n \leq 600\,000$, і так далі... У дванадцятій підзадачі $n \leq 2\,000\,000$.

Задача С. Питання

Обмеження використання часу: 1 second
Обмеження використання пам'яті: 256 megabytes

У вас є масив a довжиною n . Вам дають q запитів двох типів:

1. змінити p -те число в масиві a на v ;
2. знайти p -те число в масиві a .

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)
```

- n — довжина масиву;
- a — масив цілих чисел;
- ця функція викликається першою лише один раз. Вона потрібна для того, щоб повідомити вам розмір масиву та сам масив. Лише після виклику цієї функції, будуть викликатись інші дві.

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

- p — позиція;
- v — нове число;
- ця функція викликається, коли потрібно задати запит першого типу.

```
integer ask(integer p)
```

- p — позиція;
- ця функція викликається, коли потрібно задати запит другого типу;
- функція має повернути ціле число — відповідь на запит.

Зверніть увагу, що індексація в масивах починається з нуля, не з одиниці.

Ви можете завантажити архів за посиланням <http://ejudge.sumdu.edu.ua/statements/samples-tren.zip>.

Формат вхідних даних

Перший рядок містить два цілих числа n та m ($1 \leq n \leq 100$, $1 \leq m \leq 1000$) — довжина масиву та кількість запитів.

Другий рядок містить n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$) — числа масиву.

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

1. «1 p v » ($1 \leq p \leq n$, $1 \leq v \leq 100$) — позиція та нове число.
2. «2 p » ($1 \leq p \leq n$) — позиція.

Формат вихідних даних

Після кожного запиту другого типу буде виведена відповідь на нього.

Приклад

input	output
5 7	3
1 2 3 4 5	3
2 3	2
1 1 3	9
1 3 9	5
2 1	
2 2	
2 3	
2 5	

Оцінювання

Кожний тест, крім вхідного, оцінюється в 20 балів.

Деталі реалізації для C

Реалізуйте три функції:

```
void upd(int p, int v)
int ask(int p)
void init(int n, int *v)
```

Деталі реалізації для C++

Реалізуйте три функції:

```
void upd(int p, int v)
int ask(int p)
void init(int n, vector<int> v)
```

Деталі реалізації для Java

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

```
public static void upd(int p, int v)
public static int ask(int p)
public static void init(int n, List<Integer> a)
```

Деталі реалізації для Python

Реалізуйте три функції:

```
def upd(p, v):
def ask(p):
def init(n, a):
```

Деталі реалізації для Pascal

Реалізуйте три функції:

```
procedure upd(p, v: longint);
function ask(p: longint): longint;
procedure init(n: longint; g: array of longint);
```