



Task Languages

⌚ 0.6 sec. 💾 256 MB

In this task you will solve 10 tasks in 10 different languages for a total of 150 points. It is not allowed to use a translator. Every solution must implement all the functions and include the library **ezici.h**.



Task 1 - 27 points

Тајна меѓународна организација наречена Синдикат вработува агенти од N . Сите тие се дел од глобална разузнавачка мрежа, но неодамна меѓу нив се појавија сомнежи – кој кому му верува и кој кому се сомнева. Генералниот директор одлучи дека за да се одржи редот, агентите треба да се поделат во две групи:

- Лојалисти – оние кои веруваат во каузата
- Сомневачи – оние кои не се сигурни дали организацијата е чиста

Но, постои еден проблем:

Ако двајца агенти си веруваат еден на друг, тие мора да бидат во истата група.

Ако двајца агенти не си веруваат еден на друг, тие не можат да бидат во истата група, бидејќи ќе се шпионираат еден со друг.

Синдикатот сака да тестира дали оваа рамнотежа на доверба е можна – дали таквата поделба може да се направи без противречности.

Ако ова е можно, тогаш организацијата е стабилна.

Но, ако се открие маѓепсан круг на меѓусебно сомневање – каде што довербата и недовербата се испреплетени на таков начин што нема логично решение – тогаш системот се урива и добиваме противречност.

Напишете програма што му помага на Синдикат со проверка дали организацијата е стабилна или се јавува контрадикција.

Детали за имплементација

Треба да ја имплементирате следната функција:

```
bool task1(int N, int M, std::vector<std::pair<std::pair<int, int>, bool>> v);
```

- N – бројот на сите агенти
- M – бројот на врски меѓу сите агенти, односно колку пати е кажано дали двајца агенти си веруваат или не си веруваат еден на друг.



-
- Секој елемент во v ги означува броевите на двата агенти x, y и дали тие си веруваат еден на друг t .

Функцијата треба да врати 1 ако организацијата е стабилна, а 0 во спротивно.

Ограничувања

- $1 \leq N \leq 10^5$
- $1 \leq M \leq 2 \cdot 10^5$
- $1 \leq x, y \leq N, x \neq y$
- $0 \leq t \leq 1$



Task 2 - 12 points

Es wird ein Baum gegeben – ein ungerichteter, zusammenhängender, azyklischer Graph mit N Knoten, nummeriert von 1 bis N , und der Wurzel 1. Für jeden Knoten definieren wir die Tiefe als die Länge (in Anzahl der Kanten) des einfachen Pfads zwischen der Wurzel und diesem Knoten. Deine Aufgabe ist es, die maximale Tiefe eines Knotens im Baum zu finden.

Implementierungsdetails

Du sollst die folgende Funktion implementieren:

```
long long task2(std::vector<long long> a);
```

a – für jeden Index $0 \leq i \leq N - 2$, ist a_i der Elternknoten von Knoten $i + 2$.

Die Funktion soll das Ergebnis zurückgeben.

Einschränkungen

- $1 \leq N \leq 10^5$

Beispiel

Jury	Teilnehmer
task2({1, 1, 2, 3})	2



Task 3 - 12 points

Դուք տրված եք երկարությամբ N ամբողջ թվերի հաջորդականություն, որը կարող է պարունակել նաև բացասական թվեր: Ներկայացնում ենք հաջորդականության առավելագույն թիվը որպես այն թիվը, որը մեծ է կամ հավասար հաջորդականության յուրաքանչյուր տարրի հետ: Նույն կերպ, հաջորդականության նվազագույն թիվը ներկայացնում ենք որպես այն թիվը, որը փոքր է կամ հավասար հաջորդականության յուրաքանչյուր տարրի հետ: Ձեր առաջադրանքը գտնել առավելագույն և նվազագույն թվերի տարբերությունը հաջորդականությունում:

Կատարման մանրամասներ

Դուք պետք է իրականացնեք հետևյալ ֆունկցիան.

```
long long task3(std::vector<long long> a);
```

Ֆունկցիան ընդունում է հաջորդականությունը և վերադարձնում պատասխանը:

Պարտադիր պայմաններ

- $1 \leq N \leq 10^5$
- $-10^{18} \leq a_i \leq 10^{18}$

Օրինակ

Ձյուրի	Մրցակից
task3({1, 15, 1, -3, 15, 18, 2, 0, 14})	21



Task 4 - 12 points

Σας δίνεται μια ακολουθία από N ακέραιους αριθμούς. Ο στόχος σας είναι να μετρήσετε τον αριθμό των ζευγών i, j όπου i είναι μικρότερο από j και $a_i + a_j$ είναι άρτιος.

Λεπτομέρειες υλοποίησης

Πρέπει να υλοποιήσετε την παρακάτω συνάρτηση:

```
long long task4(std::vector<long long> a);
```

Η συνάρτηση πρέπει να επιστρέφει το αποτέλεσμα.

Περιορισμοί

- $1 \leq N \leq 10^5$
- $-10^{18} \leq a_i \leq 10^{18}$

Παράδειγμα

Κριτής	Διαγωνιζόμενος
task4({1, 15, 1, -3, 15, 18, 2, 0, 14})	16



Task 5 - 12 points

Vi se dă o secvență de N numere. Găsiți numărul de numere pare din secvența dată.

Detalii de implementare

Trebuie să implementați următoarea funcție:

```
long long task5(std::vector<long long> a);
```

Funcția trebuie să returneze rezultatul.

Constrângeri

- $1 \leq N \leq 10^5$
- $-10^{18} \leq a_i \leq 10^{18}$

Exemplu

Juriu	Competitor
task5({1, 15, 1, -3, 15, 18, 2, 0, 14})	4



Task 6 - 12 points

Gegeven is een reeks van N gehele getallen. Tel hoeveel elementen a_i strikt groter zijn dan alle voorafgaande elementen in de reeks.

Implementatiedetails

Je moet de volgende functie implementeren:

```
long long task6(std::vector<long long> a);
```

De functie moet het resultaat retourneren.

Beperkingen

- $1 \leq N \leq 10^5$
- $-10^{18} \leq a_i \leq 10^{18}$

Voorbeeld

Jury	Concurrent
<code>task6({1, 15, 1, -3, 15, 18, 2, 0, 14})</code>	3



Task 7 - 12 points

Rhoddir i chi restr o N rhif cyfan. Diffiniwn werth gwahanol fel unrhyw rif sy'n ymddangos o leiaf unwaith yn y rhestr, wedi'i gyfrif unwaith yn unig ni waeth sawl gwaith y mae'n ailadrodd. Eich tasg yw cyfrif faint o werthoedd gwahanol sydd yn y rhestr a roddwyd.

Manylion Gweithredu

Mae angen i chi weithredu'r swyddogaeth ganlynol:

```
long long task7(std::vector<long long> a);
```

Dylai'r ffwythiant ddychwelyd yr ateb.

Cyfyngiadau

- $1 \leq N \leq 10^5$
- $-10^{18} \leq a_i \leq 10^{18}$

Enghraifft

Rheithgor	Cystadleuydd
<code>task7({1, 15, 1, -3, 15, 18, 2, 0, 14})</code>	7



Task 8 - 27 points

Дадзены масіў з N цэлых лікаў, пранумараваных ад 0 да $N - 1$. Знайдзіце любую непустую падпаслядоўнасць паслядоўнасці, не абавязкова з паслядоўнымі элементамі, з сумай, якая дзеліцца на N без астатку.

Падрабязнасці рэалізацыі

Вам неабходна рэалізаваць наступную функцыю:

```
std::vector<long long> task8(std::vector<long long> a);
```

Функцыя павінна вяртаць спіс індэксаў, якія складаюць падпаслядоўнасць.

Абмежаванні

- $1 \leq N \leq 10^5$
- $0 \leq a_i \leq 10^{12}$

Прыклад

Журы	Удзельнік
task8({1, 15, 1, 3, 15, 18, 2, 0, 14})	{5}



Task 9 - 12 points

```

01000100 01100001 01100100 01100101 01101110 01101001 00100000 01110011
01100001 00100000 01001110 00100000 01100011 01100101 01101100 01101001
00100000 01100011 01101000 01101001 01110011 01101100 01100001 00101110
00100000 01001110 01100001 01101101 01100101 01110010 01100101 01110100
01100101 00100000 01101101 01100001 01101011 01110011 01101001 01101101
01100001 01101100 01101110 01100001 01110100 01100001 00100000 01110110
01110101 01111010 01101101 01101111 01101010 01101110 01100001 00100000
01110011 01110101 01101101 01100001 00100000 01101110 01100001 00100000
01110000 01101111 01110011 01101100 01100101 01100100 01101111 01110110
01100001 01110100 01100101 01101100 01100101 01101110 00100000 01110000
01101111 01100100 01101101 01100001 01110011 01101001 01110110 00101110
  
```

```

01001001 01101101 01110000 01101100 01100101 01101101 01100101
01101110 01110100 01100001 01100011 01101001 01101111 01101110
01101110 01101001 00100000 01100100 01100101 01110100 01100001
01101001 01101100 01101001
  
```

```

01010100 01110010 01110001 01100010 01110110 01100001 00100000 01100100
01100001 00100000 01101001 01101101 01110000 01101100 01100101 01101101
01100101 01101110 01110100 01101001 01110010 01100001 01110100 01100101
00100000 01110011 01101100 01100101 01100100 01101110 01100001 01110100
01100001 00100000 01100110 01110101 01101110 01101011 01100011 01101001
01110001 00111010:
  
```

```
long long task9(std::vector<long long> a);
```

```

01000110 01110101 01101110 01101011 01100011 01101001 01110001 01110100
01100001 00100000 01110100 01110010 01110001 01100010 01110110 01100001
00100000 01100100 01100001 00100000 01110110 01110101 01110010 01101110
01100101 00100000 01110100 01110101 01110010 01110011 01100101 01101110
01100001 01110100 01100001 00100000 01110011 01110101 01101101 01100001
00101110.
  
```

```

01001111 01100111 01110010 01100001 01101110 01101001 01100011
01101000 01100101 01101110 01101001 01110001
  
```

- $1 \leq N \leq 00000001\ 10000110\ 10100000$
- $a_i \geq -11101000\ 11010100\ 10100101\ 00010000\ 00000000$
- $a_i \leq 11101000\ 11010100\ 10100101\ 00010000\ 00000000$

Task Languages

🌐 Macedonian, German, Armenian, Greek,
Romanian, Dutch, Welsh, Belarusian,
Binary numeral system, Old Church
Slavonic



01010000 01110010 01101001 01101101 01100101 01110010

01001010 01110101 01110010 01101001	01010011 01110101 01110011 01110100 01100101 01111010 01100001 01110100 01100101 01101100
task9({1, 15, 1, -3, 15, 18, 2, 0, 14})	63



Task 10 - 12 points

Дана сѣтъ N цѣла числа. Обрѣтѣте цѣнѣ сложения по бити исключающааго всѣхъ чиселъ тѣхъ.

Сѣдѣланиѣ

Подобаѣтъ вамъ сѣдѣлати дѣло сеѣ:

```
long long task10(std::vector<long long> a);
```

Дѣло то отъвѣтъ да въздастъ.

Прѣдѣли

- $1 \leq N \leq 10^5$
- $0 \leq a_i \leq 10^{18}$

Образъ

Зѣваниѣ	Отъвѣтъ
<code>task10({1, 15, 1, 3, 15, 18, 2, 0, 14})</code>	29



Sample grader

You are provided with a sample grader `Lgrader.cpp`.

The format of the input is the following:

- The first line contains the numbers N , M , and the next M lines contain three numbers each, respectively the values of every element in v .
- The next line contains N and then $N - 1$ numbers, as described in task two.
- Each of the next eight lines contains N and then N numbers - the input of the respective task (from the third to the tenth).

On ten lines of the standard output the grader prints the answers returned by the solution.