# II (районний) етап Всеукраїнської учнівської олімпіади з інформатики

**Київ, 2019/20 н. р.**

---

Максимальна оцінка за кожну з чотирьох задач — 100 балів.
Для всіх задач обмеження на час — 1 секунда / тест; обмеження на пам'ять — 256 МБ.

Матеріали олімпіади буде оприлюднено на сайті **kievoi.ippo.kubg.edu.ua**, а також на **soi.org.ua**.
Автор задач — Данило Мисак.

---

## 1. Комікси (назва програми: `comics.cpp` / `comics.pas` / `comics.*`)

Ярослав і Мирослава колекціонують комікси. Ярослав має $a$ коробок по $n$ коміксів у кожній та $b$ коробок по $m$ коміксів у кожній, а Мирослава — $a$ коробок по $m$ коміксів у кожній та $b$ коробок по $n$ коміксів у кожній. У кого з дітей коміксів більше?

**Вхідні дані**

У вхідному файлі через пробіл вказано чотири числа: $a$, $b$, $n$ та $m$ (саме в такому порядку). Усі чотири числа натуральні та не перевищують $10\,000$.

**Вихідні дані**

У вихідний файл виведіть `1`, якщо більшу кількість коміксів має Ярослав; `2`, якщо більше коміксів у Мирослави; `0`, якщо Ярослав та Мирослава мають однакову кількість коміксів.

**Приклади**

| Вхідний файл `comics.in` | Вихідний файл `comics.out` |
|---|---|
| `2 3 5 4` | `2` |

**Коментар до прикладу**

Ярослав має $2 \times 5 + 3 \times 4 = 22$ комікси, а Мирослава — $2 \times 4 + 3 \times 5 = 23$ комікси, тобто на один комікс більше, ніж Ярослав.

## 2. Монети (назва програми: `coins.cpp` / `coins.pas` / `coins.*`)

Ярослав та Мирослава мають спільну колекцію з $n$ монет. Як символ своєї дружби вони хочуть окремо зберігати таку пару монет, що в сумі номінальна вартість цих двох монет дає особливе число $s$. Підрахуйте кількість різних способів вибрати потрібну пару.

**Вхідні дані**

У першому рядку вхідного файлу вказано натуральні числа $s$ та $n$, не менші за $2$. У другому рядку записано $n$ натуральних чисел — номінальні вартості монет із колекції. Усі числа у вхідному файлі (включно з числами $s$ та $n$) не перевищують $200\,000$.

**Вихідні дані**

У вихідний файл виведіть єдине число — кількість способів вибрати дві монети з сумарною номінальною вартістю $s$. Відомо, що шукана кількість не перевищує $10^9$.

**Приклади**

| Вхідний файл **coins.in** | Вихідний файл **coins.out** |
|---|---|
| 4 5<br>2 2 3 2 1 | 4 |
| 10 3<br>6 2 10 | 0 |

**Коментарі до прикладів**

У першому прикладі діти можуть вибрати пару в один із чотирьох способів: взяти першу і другу монети; або першу й четверту; або другу й четверту; або третю та п'яту.

У другому прикладі жодні дві монети, на жаль, не дають у сумі вартість $10$.

## 3. Марки (назва програми: **stamps.cpp** / **stamps.pas** / **stamps.***)

Нещодавно на уроці математики Ярослав і Мирослава вивчили, що арифметичною прогресією називають послідовність чисел, у якій різниця між кожними двома сусідніми членами однакова. А невдовзі після того діти дізналися, що на честь ювілею математичного товариства столиці було випущено дві серії марок. Кожна серія складається з $n$ марок різної номінальної вартості, і ці $n$ номіналів утворюють арифметичну прогресію. Для своєї колекції марок Ярослав придбав одну з цих серій, а Мирослава — іншу. Однак, роздивляючись придбання одне одного, діти ненароком перемішали всі марки.

Знаючи номінали марок — $2n$ попарно різних чисел, — допоможіть дітям розділити марки на дві серії. Відомо, що це можна зробити рівно в один спосіб.

**Вхідні дані**

У першому рядку вхідного файлу вказано натуральне число $n$ — кількість марок у серії, $3 \leqslant n \leqslant 100\,000$. У другому рядку записано $2n$ різних натуральних чисел, менших за $10^9$, — перемішані номінали марок.

**Вихідні дані**

У перший рядок вихідного файлу виведіть в порядку зростання всі номінали марок Ярославової серії, а в другий рядок — усі номінали марок Мирославиної серії (так само в порядку зростання). Діти пам'ятають, що найдешевша марка Ярослава має менший номінал, ніж найдешевша марка Мирослави.

**Приклад**

| Вхідний файл **stamps.in** | Вихідний файл **stamps.out** |
|---|---|
| 4<br>7 9 23 3 16 15 11 2 | 2 9 16 23<br>3 7 11 15 |

**Коментар до прикладу**

Виведені у вихідний файл послідовності утворюють шукані серії марок, адже є арифметичними прогресіями: $9 - 2 = 16 - 9 = 23 - 16$ та $7 - 3 = 11 - 7 = 15 - 11$. Серії виведено в правильному порядку, бо $2 < 3$.

## 4. Фантики (назва програми: **wrappers.cpp** / **wrappers.pas** / **wrappers.***)

За останній час Ярослав і Мирослава назбирали разом $n$ фантиків. Як відомо, кожен колекціонер прагне, щоб його колекція була максимально розмаїтою. Тому діти хочуть розподілити між собою фантики так, щоб

жодні два чимось схожих між собою фантики не потрапили до одного власника. Для цього Ярослав та Мирослава занумерували фантики числами від $1$ до $n$ та виписали, які саме пари фантиків виглядають подібно. Усього в них вийшло $m$ пар, причому номери деяких фантиків могли бути виписані в кількох різних парах.

Допоможіть дітям розподілити фантики бажаним чином або визначте, що це неможливо.

**Вхідні дані**

У першому рядку вхідного файлу вказано два натуральних числа $n$ та $m$ — кількість фантиків та кількість їх подібних пар; $2 \leqslant n < 2000$, $1 \leqslant m < 450\,000$. У кожному з наступних $m$ рядків задано по два числа $a_i$ та $b_i$ — номери схожих між собою фантиків, $1 \leqslant a_i < b_i \leqslant n$, $1 \leqslant i \leqslant m$. Жодна пара номерів $a_i$, $b_i$ у вхідному файлі не повторюється. Крім того, вхідні дані гарантують, що є не більше ніж один спосіб розподілити фантики між дітьми, щоб жодні два схожих між собою фантики не опинилися в одного власника.

**Вихідні дані**

У першому рядку вихідного файлу виведіть у порядку зростання номери фантиків, які мають опинитися в того ж власника, що й фантик під номером $1$ (включно з самим числом $1$). У другому рядку виведіть у порядку зростання номери фантиків, які повинні опинитися в іншого власника.

Якщо жоден розподіл фантиків не задовольняє умову задачі, в обох рядках виведіть по нулю.

**Приклади**

| Вхідний файл **wrappers.in** | Вихідний файл **wrappers.out** |
|---|---|
| 5 6<br>2 5<br>2 3<br>1 5<br>4 5<br>3 4<br>1 3 | 1 2 4<br>3 5 |
| 3 3<br>1 2<br>2 3<br>1 3 | 0<br>0 |

# Ідеї розв'язання

## 1. Комікси

Слід порівняти два числа $an + bm$ та $am + bn$. Якщо більшим є перше число, слід вивести `1`; якщо друге число — `2`; якщо числа рівні, треба вивести `0`. Слід також врахувати, що числа можуть вийти за межі двобайтових змінних і використати відповідний тип даних.

## 2. Монети

Безпосередній перебір принесе лише частковий бал, бо для великих $n$ програма не встигне за відведений час (секунду) перебрати всі можливі пари чисел. Замість цього можна скористатися іншим підходом. Для кожного числа від $1$ до $200\,000$ слід підрахувати, скільки монет мають дану номінальну вартість: завести масив $cnt$, у комірці $cnt[p]$ якого буде зберігатися кількість монет вартості $p$ ($1 \leqslant p \leqslant 200\,000$), і після зчитування кожного наступного числа збільшувати відповідну комірку масиву на $1$. Далі через $\left[\frac{s-1}{2}\right]$ ми позначимо найбільше ціле число, що не перевищує $\frac{s-1}{2}$. Для всіх значень $p$ у межах від $1$ до $\left[\frac{s-1}{2}\right]$ включно добуток $cnt[p] \times cnt[s-p]$ дорівнюватиме кількості пар монет, одна з яких має вартість $p$, а інша — вартість $s-p$; при цьому $p < s-p$. Сума всіх $\left[\frac{s-1}{2}\right]$ таких добутків буде кількістю варіантів вибрати пару монет із сумарною вартістю $s$ за умови, що вартості цих двох монет різні. Залишається врахувати пари монет однакової вартості: якщо $s$ непарне, таких пар немає, а інакше їх рівно $\frac{cnt[s/2]\cdot(cnt[s/2]-1)}{2}$.

## 3. Марки

Один з можливих способів розв'язати задачу такий. Відсортуємо всі $2n$ чисел у порядку від найменшого до найбільшого. Серед трьох найменших елементів відсортованої послідовності принаймні два належатимуть до однієї й тієї ж прогресії, причому будуть двома найменшими її членами (а якщо всі три найменші числа належать одній і тій самій прогресії, то найменшими двома її членами є, очевидно, два перших числа). Тепер, послідовно розглядаючи гіпотези про те, що двома найменшими членами однієї з прогресій є перший і другий; перший і третій; другий і третій елементи відсортованої послідовності, встановимо, котра з цих гіпотез є правильною. Для перевірки можемо скористатися таким підходом: перебиратимемо в порядку збільшення всі $2n$ чисел; якщо чергове число, яке ми розглядаємо, є таким, що підходить до першої прогресії (а, беручи припущення гіпотези, ми вже знаємо і перший член цієї прогресії, і її різницю, і кількість елементів), то долучаємо це число до першої прогресії, інакше — до другої. Якщо обидві побудовані послідовності дійсно є арифметичними прогресіями (з $n$ елементами в кожній), то маємо відповідь; інакше переходимо до наступної гіпотези. Описаний процес перевірки можна втілити за один лінійний прохід послідовності.

Оскільки алгоритм складається з сортування і кількох лінійних проходів масиву на $2n$ елементів, його складність можна оцінити як $O(2n \log 2n) = O(n \log n)$. Це дозволяє заробити повний бал.

Утім, алгоритм можна оптимізувати і до лінійного. Для пошуку трьох найменших елементів замість сортування використаємо, наприклад, три послідовних лінійних проходи. А для перевірки гіпотези перебиратимемо числа не в порядку збільшення, а в довільному. Це не завадить відібрати з набору ті й лише ті числа, що належать до першої прогресії. Щоб установити, чи решта чисел утворюють другу прогресію, знайдемо шляхом двох лінійних проходжень два найменших числа, що не потрапили до першої прогресії: вони якраз і мають становити два перших члени другої прогресії. Залишається ще раз пройтися по масиву і перевірити, чи решта чисел у ньому «узгоджуються» зі знайденими першими членами потенційної прогресії. При цьому, звичайно, не слід забувати, що в обох прогресіях повинно бути рівно по $n$ членів. Нарешті, щоб уникнути сортування при виведенні відповіді, можна конструювати прогресії безпосередньо з їхніх арифметичних властивостей (а не виводити у вихідний файл упорядковані елементи масиву).

## 4. Фантики

Задачу можна переформулювати в термінах теорії графів. Нехай фантики — це вершини графа, а ребро між двома вершинами проведене тоді й лише тоді, коли два відповідних фантики схожі. Нам необхідно розділити всі вершини графа на дві частини так, щоб кожне ребро графа сполучало вершини, що належать до різних частин. Граф, для якого це вдається зробити, називається дводольним, а відповідні його частини — долями.

Задачу можна розв'язати з допомогою пошуку в глибину або в ширину. Присвоївши вершині $1$ власника № $1$ і запустивши будь-який із типів пошуку з цієї вершини, будемо кожній новій пройденій вершині графа присвоювати власника № $2$, якщо ми прийшли у неї з вершини, якій було присвоєно власника № $1$, і навпаки. При цьому якщо дана вершина сполучена ребром хоча б з однією іншою вершиною, якій раніше присвоїли того самого власника, що й даній вершині, то розбити граф на дві частини неможливо.

Після завершення пошуку в глибину або в ширину, якщо вдалося присвоїти власників всім вершинам, маємо потрібний розподіл вершин у компоненті зв'язності графа, яка містить вершину $1$, а інакше виводимо нулі. Якщо граф складається з єдиної компоненти зв'язності, виводимо знайдений розподіл. Якщо ж у графі є дві або більше компонент зв'язності, тобто якщо після завершення пошуку, запущеного з вершини $1$, одна чи кілька вершин залишились невідвіданими, то слід також вивести нулі. Справді: якщо потрібний розподіл вершин усіх компонент зв'язності існує, то він не єдиний, адже розподіли різних компонент можна як завгодно комбінувати. А згідно з умовою задачі якби розподіл існував, то він мав би бути єдиним. Отже, у випадку двох і більше компонент зв'язності умова задачі гарантує відсутність потрібного розподілу.

Насамкінець зауважимо, що обмеження $m < 450\,000$ має суто технічний характер і пов'язане з необхідністю обмежити розмір вхідного файлу.