## 4. Рядковий автомат (Данило Мисак)

Зауважимо, що, як і в більшості інших задач, подані тут ідеї не є єдино правильним підходом до відповідних підзадач. Усі або майже всі підзадачі мають багато різних розв’язків.

**Підзадача 1.** Слід узяти всі пари сусідніх цифр, що йдуть у порядку зменшення, та поміняти їх порядок на протилежний: 10 міняємо на 01, 20 на 02, ..., 98 — на 89. Порядок, у якому записано правила, ролі не грає.

**Підзадача 2.** Заміняємо блоки по 10 літер I на деяку іншу літеру: скажімо, J. Після таких замін у кінці рядка залишиться така кількість літер I, що дорівнює останній цифрі шуканого числа. Відповідно, заміняємо дев’ять літер I на дев’ятку, вісім — на вісімку, ..., одну — на одиницю. Отримали правильну останню цифру, зліва від якої кількість літер J дорівнює кількості повних десятків у шуканому числі. Тоді можна з літерами J провести ту саму операцію: замінити блоки з 10 таких літер, скажімо, на K та «згорнути» ті літери J, що залишилися, у цифру. Ця цифра і є розрядом десятків шуканого числа. Залишається «згорнути» сотні: їх (тобто літер K) може залишитися не більше за 9.

**Підзадача 3.** Дещо модифікуємо попередній алгоритм: замінятимемо блоки по 10 літер І на J0, після чого відразу 0J та 0I — на J та I відповідно. Таким чином дістанемо лише один нуль у кінці рядка, причому тоді й лише тоді, коли всі літери I розбилися на групи по 10 без залишку. Крім того, як і раніше, дев’ять I замінимо на дев’ятку, вісім I на вісімку і т. д. Далі десять J замінимо на K0, проведемо аналогічні операції з J і K. Нарешті, десять K замінимо на десятку, дев’ять K на дев’ятку, ..., одну K — на одиницю.

**Підзадача 4.** Замінимо цифру 0 на A, 1 — на AI, 2 — на AII, ..., 9 — на A і дев’ять літер I. Тепер якщо після деякої літери I стоїть у сумі літер A, то дана літера позначає розряд . Тому, замінюючи послідовно фрагмент IA на A і десять літер I, у підсумку отримаємо якраз таку кількість літер I, яка дорівнює початковому числу[[1]](#footnote-1). При цьому зліва від них стоятимуть усі наявні літери A. Залишається замінити A на порожній рядок. Єдине: замінити IA на A і десять I нам не дозволяють обмеження на розмір рядка заміни. Але з цієї ситуації нескладно вийти, замінивши натомість IA на AB, а першим правилом у програмі — B на десять I.

**Підзадача 5.** Спочатку встановимо індикатор початку рядка: замінимо всі a на 0a, b на 0b, c на 0c, а всі a0 на a, b0 на b, а c0 — на c. Після таких операцій у нас залишиться єдиний нуль на першій позиції[[2]](#footnote-2). Тепер будемо робити великими літери, що опинилися на початку рядку, тобто замінювати 0a на 0A, 0b на 0B, 0c на 0C. Водночас будемо переставляти всі 9 пар з великих та малих літер, коли велика стоїть перед малою. Так ми спочатку пересунемо першу літеру в кінець, потім друга літера стане на передостаннє місце і т. д. У кінці залишається замінити 0 на порожній рядок. Єдина проблема тут така, що при початковій зміні, наприклад, символу a на 0a послідовність операцій буде зациклюватися. Тому мінятимемо a на 0x, b на 0y, c на 0z, після чого ототожнюватимемо x із a, y із b, а z — із c.

**Підзадачі 6 і 7.** Ідея базується на розкладі обох доданків у відповідну кількість літер I, після чого літери (яких тепер якраз сумарна кількість) збираються назад у число. Основна проблема полягає в тому, що після того, як ми зберемо остаточне число, рядковий автомат знов захоче розкладати числа в літери I. Щоб цього не сталося, ми маємо використати єдину відмінність між вхідними та потенційними вихідними даними: наявність плюса. Локальність цього плюса можна побороти таким чином: замінюватимемо, скажімо, не 4 на AIIII, а 4+ на pAIIII+ та аналогічно +4 на +AIIIIp. Тоді за допомогою символу p ми зможемо пропагувати зміни (а точніше, уникати зайвих змін) також і значно лівіше та правіше від плюса. Після цього плюс та символ p прибираються та йде зворотний процес. Проблеми з розміром рядка заміни вирішуємо так само, як і в пункті 4. Цілком імовірно, що кількість виконуваних операцій при цьому вийде за допустиму межу. Тоді можна оптимізувати програму: додатково до заміни IA на AB (див. четверту підзадачу) введемо заміни IIA на ABB, IIIA на ABBB і т. д. Щоб перша частина алгоритму (зведення до одиниць) не конфліктувала з другою (зведення до десяткового запису), можемо в першій частині за одиниці брати літери I, далі міняти їх на J, а в другій частині одиницями вважати вже літери J, а I ніяк не зачіпати. Щоб оптимізувати кількість операцій, потрібно ввести не лише заміну I на J, але й, наприклад, II на JJ, III на JJJ і т. д. Насамкінець зауважимо, що в третьому пункті ми працювали лише з числами в межах до 1000, а в даній підзадачі сума може вийти більшою. Отже, потрібно доповнити відповідні правила для коректної роботи з числами до 2000.

1. Але див. зауваження нижче. [↑](#footnote-ref-1)
2. Але див. зауваження нижче. [↑](#footnote-ref-2)