

Олімпіада з інформатики Солом'янського району

2011/12 н. р.

Автор задач — Данило Мисак

1. Подарунок (назва програми: `present.pas` / `present.cpp` / `present.dpr`)

Якось за успіхи в олімпіадах з інформатики вчителі вирішили зробити Петрику П'яточкіну подарунок. Вони запропонували йому вибрати один із двох подарункових наборів: взяти a ручок та b олівців або, замість цього, взяти a олівців та b ручок. Відомо, що одна ручка коштує x гривень, а один олівець — y гривень. Визначте, який із наборів вигідніше взяти Петрикові.

Вхідні дані

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

Вихідні дані

У вихідний файл виведіть 1, якщо Петрикові вигідніше взяти a ручок та b олівців; 2, якщо хлопцю краще взяти a олівців та b ручок; 0, якщо обидва варіанти однаково вигідні.

Приклад

Вхідний файл <code>present.in</code>	Вихідний файл <code>present.out</code>
2 3 4 3	2

2. Шифр (назва програми: `cipher.pas` / `cipher.cpp` / `cipher.dpr`)

Якось під час прогулянки Петрик П'яточкін придумав геніальну схему шифрування чисел. Згідно з Петриковою схемою, спершу всі цифри числа слід записати по колу. Далі з усіма цифрами водночас виконують таку операцію: замість цифри записують останню цифру суми цієї цифри та двох сусідніх до неї цифр. Далі коло «розривають» у тому ж місці, в якому «склеювали» початкове число. Отримане число і є шифром.

Перед вами стоїть задача із розшифрування Петрикового кодування: знаючи зашифроване число, відновіть його початкове значення. Під числом у цій задачі розуміємо послідовність цифр від 0 до 9, при цьому на початку числа може стояти один або кілька нулів.

Вхідні дані

У першому рядку вхідного файлу записано кількість цифр n Петрикового шифру. Другий рядок файлу містить число, що є результатом Петрикового шифрування. Цифри числа розділені пробілами. Кількість n цифр у числі — від 3 до 100 000. Відомо, що в половині тестів n не перевищує 6.

Вихідні дані

У вихідний файл виведіть число, яке зашифрував Петрик, не розділяючи його цифри пробілами. Якщо однозначно визначити зашифроване Петриком число неможливо або якщо Петрик помилився і жодне число не підходить під його шифр, виведіть у вихідний файл число 0.

Приклади

Вхідний файл <code>cipher.in</code>	Вихідний файл <code>cipher.out</code>
4 5 3 2 0	8057
3 3 3 3	0

Пояснення до прикладів

Розгляньмо **перший приклад**. Якщо записати цифри 8057 по колу, сусідніми цифрами із вісімкою будуть 0 і 7, а із сімкою — 5 та 8. Тоді сума цифри 8 і сусідніх з нею складе $8 + 0 + 7 = 15$, і остання цифра цієї суми 5 якраз стоїть у шифрі на першому місці. Сума другої цифри числа, — нуля, — і сусідніх з нею складає $0 + 8 + 5 = 13$, і остання цифра цієї суми 3 якраз стоїть у шифрі на другому місці. Аналогічно, цифра 2 — остання цифра суми $5 + 0 + 7 = 12$ — стоїть на третьому місці, а 0 — остання цифра суми $7 + 5 + 8 = 20$ — на четвертому.

Розгляньмо **другий приклад**. Шифр 333 відповідатиме відразу кільком числам, зокрема числам 111 і 760: $1 + 1 + 1 = 3$; $7 + 6 + 0 = 6 + 0 + 7 = 0 + 7 + 6 = 13$. Тож у вихідний файл слід вивести 0.

3. Гноми (назва програми: `gnomes.pas` / `gnomes.cpp` / `gnomes.dpr`)

Якось після прогулянки Петрику П'яточкіну приснився дивний сон. Хлопцю наснилися n гномів, що стоять у шерензі. Кожен із них мав (можливо, нульову) кількість цукерок. Щосекунди всі гноми, крім останнього (найправішого), водночас робили таке: перевіряли, скільки в них є цукерок, і якщо кількість цукерок у гнома була більшою за a , але не перевищувала b , він віддавав одну свою цукерку сусіду, який стояв справа (якщо дивитися на шеренгу спереду). Через певний час ніхто з гномів цукерками більше не ділився, бо в усіх гномів, крім, можливо, останнього, кількість цукерок або не перевищувала a , або була більшою за b . Знаючи, скільки цукерок було в кожного гнома спочатку, з'ясуйте, скільки кожен гном мав цукерок після того, як передавання солодоців припинилося.

Вхідні дані

Вхідний файл складається із двох рядків. У першому рядку вказано числа n , a та b (саме в такому порядку), а в другому рядку через пробіл задано початкові кількості цукерок у гномів у порядку від лівого гнома до правого. Всі числа у вхідному файлі цілі й лежать у межах від 0 до 10 000 включно. Кількість гномів n ненульова.

Вихідні дані

Єдиний рядок вихідного файла повинен містити n чисел, розділених пробілом: кінцеві кількості цукерок у гномів у порядку від лівого гнома до правого.

Приклад

Вхідний файл <code>gnomes.in</code>	Вихідний файл <code>gnomes.out</code>
5 1 4 5 3 2 0 1	5 1 1 1 3

Пояснення до прикладу

Сек.	Розташування цукерок	Коментар
0	5 3 2 0 1	Перший гном має більше ніж 4 цукерки, тому ні з ким ними не ділиться. Другий, третій та четвертий гноми діляться своїми цукерками, якщо вони їх мають на даний момент хоча б дві. Останній гном тільки збирає цукерки й нікому свої не віддає.
1	5 2 2 1 1	
2	5 1 2 2 1	
3	5 1 1 2 2	
4	5 1 1 1 3	Коли цукерки розмістилися в кількостях 5 1 1 1 3, вони перестали рухатися, бо в жодного гнома, крім останнього, кількість цукерок не лежить у межах від 2 до 4.

Ідеї розв'язання

Подарунок

Слід порівняти два числа $ax + by$ та $ay + bx$. 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.

Шифр

Можна показати, що зашифроване число відновлюється, причому однозначно, тоді й лише тоді, коли кількість n цифр у числі (і в шифрі) не ділиться на 3.

Знайти саме зашифроване число можна так. Перебираємо перші дві цифри відповіді, із них однозначно відновлюємо решту цифр, використовуючи другу, третю, ..., передостанню цифри шифру. Якщо при цьому перша та остання цифри шифру теж відповідають знайденому числу, воно є шуканим.

Можна зробити й по-іншому: фактично, розв'язати на комп'ютері систему з n рівнянь за модулем 10, які випливають з умови.

Гноми

Задача розв'язується лінійним проходом масиву від першого до останнього гнома. На кожному кроці запам'ятовується загальна кількість перенесених (за увесь час) від попередніх гномів до поточного цукерок. Нехай у поточного k -го гнома спочатку було x_k цукерок, а передали йому в сумі y_k цукерок. Тоді можливі кілька випадків:

- Якщо $x_k > b$, то в k -го гнома наприкінці опиниться $x_k + y_k$ цукерок, і жодної цукерки він не передасть.
- Якщо $a - y_k < x_k \leq b$, то у гнома наприкінці опиниться a цукерок, а передасть праворуч він рівно $x_k + y_k - a$ одиниць солодкого товару.
- Якщо $x_k \leq a - y_k$, то, як і в першому випадку, у гнома наприкінці буде $x_k + y_k$ цукерок, і жодної цукерки він не передасть.

Останнього гнома слід розглянути окремо. В нього завжди опиняться всі цукерки, які передали йому зліва, разом із тими, які він мав від початку.

Задачу можна розв'язувати, симулюючи рух цукерок згідно з описаними в умові правилами, що дасть частковий бал.