



Двоякі числа

Натуральне число називається двояким, якщо в його десятковому запису зустрічається не більше двох різних цифр.

1. Створіть програму, що знаходить двояке число, яке найменше відрізняється від

заданого натурального числа

N за модулем.

Вхідні дані

Значення **N** - натуральне число, що не перевищує 2^{32} , вводиться з клавіатури або задається як константа.

Вихідні дані (виводяться на екран чи зберігаються у файл)

Значення **M** - найближче до **N** дwoяке число (якщо знайдеться два таких числа, то **M** - найменше з них), виводиться на екран.

2. Заповніть тестову таблицю за результатами роботи програми:

Номер

тесту

N
M

1	123	122
2	51	
3	7777	
4	1299	
5	9867	
6	9043	
7	9050	
8	10632	
9	26354	
10	998877	

Аналіз розв'язку задачі «Двоякі числа»

Для вирішення задачі скористаємося методом послідовного перебору. Будемо перевіряти числа, починаючи з **N** і поступово віддаляючись від **N**: спочатку такі, що відрізняються від **N** на 1 (менше на -1, більше на +1), потім на 2 (менше на -2, більше на +2), потім на 3 і т.д.

Введемо допоміжні змінні:

M - поточне число, яке піддається перевірці;

модуль різниці між **M** і **N** зберігатимемо у змінній **d**, знак різниці - у змінній *sign*, яка набуватиме значень -1 (

M
N
);

c[1] і **c[2]** - змінні для зберігання двох різних цифр двоякого числа;

k - кількість знайдених різних цифр у числі **M**;

a - допоміжна змінна, використовувана для виділення цифр числа **M**;

F - змінна-прапорець, яка зберігає значення 0, доки двояке число не знайдене, і

набуває значення 1, якщо таке число знайдене.

Шукане значення M можна знайти за допомогою такого алгоритму:

{надаємо початкових значень змінним; враховуємо, що числа, які менші 100, є двоякими}

M := N;

якщо M **то** F := 1

інакше

початок

F := 0;

d := 0;

sign := 1;

кінець

{розпочинаємо цикл перебору чисел до знаходження двоякого числа}

повторювати

M := N + sign * d;

a := M;

k := 0;

{перевіряємо в циклі, чи є поточне число M (тобто a) двояким}

поки k 0

початок циклу

r := a mod 10;

якщо (k=0) or (k=1) and (r c[1]) **то**

початок

k := k+1;

c[k] := r;

кінець

якщо (k=2) and (r c[1]) and (r c[2]) **то** k := k+1;

a := a div 10;

кінець циклу;

якщо k