

Завдання № 5. Порахувати кількість одиничних бітів числа

Формулювання. Дано натуральне число менше 16. Порахувати кількість його одиничних бітів. Наприклад, якщо дано число 9, запис якого в двійковій системі числення рівною 1001_2 (підрядкові цифри 2 праворуч від числа означає, що воно записано в двійковій системі числення), то кількість його одиничних бітів дорівнює 2.

Рішення. Нам необхідна змінна для введення з клавіатури. Позначимо її як n . Так як ми повинні накопичувати кількість знайдених бітів, то виникає потреба в ще одній змінній. Позначимо її як **count** («*count*» в перекладі з англ. Означає «вважати», «підрахунок» і т. Д.). Змінні візьмемо типу **byte** (вони можуть набувати значень від 0 до 255), і нехай в даному випадку такий обсяг надлишковий, але це не принципово важливо.

Як же порахувати кількість бітів у введеному числі? Адже число ж вводиться в десятковій системі числення, і його потрібно переводити в двійкову?

Насправді все набагато простіше. Тут нам допоможе одне цікаве правило:

Залишок від ділення будь-якого числа x в системі числення з основою p на саме число p дає нам розряд одиниць числа x (його крайній розряд праворуч).

Тобто, в десятковій системі числення ми отримуємо розряд одиниць деякого числа, взявши залишок від ділення цього числа на 10. Візьмемо, наприклад, число 3468. Якщо залишок від ділення на 10 дорівнює 8, тобто розряду одиниць цього числа.

Такі ж правила панують і в арифметиці в інших системах числення, і в тому числі в двійковій системі. Пропоную поекспериментувати: запишіть на папері десяткове число, потім, використовуючи будь-який калькулятор з функцією перекладу з однієї системи числення в іншу, переведіть це число в двійкову систему числення і також запишіть результат. Потім розділіть вихідне число на 2 і знову переведіть в двійкову систему. Як воно змінилося в результаті? Цілком очевидно, що у нього пропав крайній розряд праворуч, або, як ми вже говорили раніше, розряд одиниць.

Але як це використовувати для вирішення завдання? Скористаємося тим, що в двійковій запису числа немає цифр, крім 0 і 1. Легко переконатися в тому, що склавши всі розряди двійкового числа, ми отримуємо як раз таку кількість його одиничних бітів. Це означає, що замість перевірки значень розрядів двійкового представлення числа ми можемо додавати до лічильника самі ці розряди - якщо в розряді був 0, значення лічильника не зміниться, а якщо 1, то підвищиться на одиницю.

Тепер, резюмуючи вищенаведений підсумок, можна поетапно сформулювати сам алгоритм:

1) Вводимо число n ;

2) Обнуляємо лічильник розрядів **count**. Це робиться тому, що значення всіх змінних при запуску програми вважаються невизначеними, і хоча в більшості компіляторів **Pascal** вони обнуляються при запуску, все ж вважається ознакою «хорошого тону» в програмуванні обнулити значення змінної, яка буде змінюватися в процесі роботи без попереднього привласнення їй якогось - або значення.

3) Додаємо до **count** розряд одиниць в двійковій запису числа n , тобто залишок від ділення n на 2:

```
count := count + n mod 2;
```

Строго кажучи, ми могли б не додавати попереднє значення змінної **count** до залишку від ділення, так як воно все одно було нульовим. Але ми вчинили так для того, щоб зробити код більш однорідним, далі це буде видно. Врахувавши розряд одиниць в двійковій запису n , ми повинні відкинути його, щоб досліджувати трохи нижче. Для цього розділимо n на 2. Мовою **Pascal** це буде виглядати так:

```
n := n div 2;
```

4) Тепер нам потрібно ще два рази повторити **п. 3**, після чого залишиться єдиний двійковий розряд числа n , який можна просто додати до лічильника без будь-яких доповнень:

```
count := count + n;
```

5) В результаті в змінній **count** буде зберігатися кількість одиничних розрядів в двійковій запису вихідного числа. Залишилося лише вивести її на екран.