

### Количество вхождений цифры. 10-11 класс

Составьте программу, которая принимает на вход в первой строке восьмеричную цифру  $D$  – одну из  $\{0, 1, 2, 3, 4, 5, 6, 7\}$ , во второй строке целое положительное число  $L$  и в третьей строке целое неотрицательное число  $N$ , записанное в шестнадцатеричной системе. В этой записи используются десятичные цифры и заглавные латинские буквы  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ . Число  $L$  – это длина шестнадцатеричной записи числа  $N$ , и оно не более чем 49152. Программа находит, количество вхождений цифры  $D$  в запись числа  $N$ , если его перевести в восьмеричную систему. В начале записи числа  $N$  могут стоять незначащие нули, которые не следует учитывать при подсчёте количества вхождений  $D = 0$ . Незначащим является любой ноль, стоящий левее первой ненулевой цифры, или, если  $N = 0$ , то все нули, кроме самого правого.

*Формат ввода:* В первой строке содержится символ  $D$  – восьмеричная цифра (одна из  $\{0, \dots, 7\}$ ). Во второй строке содержится целое положительное число  $L$  – длина записи числа  $N$  в шестнадцатеричной системе, в которой могут быть незначащие нули ( $0 < L < 49152$ ). В третьей строке содержится непустая последовательность символов, являющаяся записью в шестнадцатеричной системе числа  $N$  – целого, неотрицательного (в этой записи  $L$  символов). В записи числа  $N$  используются десятичные цифры и заглавные латинские буквы  $\{0, \dots, 9, A, \dots, F\}$ .

*Формат вывода:* В первой и единственной строке выводится неотрицательное целое число, равное искомому количеству вхождений цифры  $D$  в восьмеричную запись числа  $N$  (от 0 до 65535).

*Пример №1:*

*ввод:*

7

2

FF

*вывод:*

2

*Пример №2:*

*ввод:*

0

32

00000000000000000000000000000000

*вывод:*

1

*Пример №3:*

*ввод:*

0

21

2020202020202020202

*вывод:*

17

### Решение

В решении нужно в цикле последовательно считывать и обрабатывать введённые цифры шестнадцатеричной записи числа. Для перевода в восьмеричную систему из шестнадцатеричной можно использовать тот факт, что  $2^4 = 16$  и  $2^3 = 8$ . Значит, тройку подряд идущих шестнадцатеричных цифр можно переводить в четвёрку восьмеричных. Заранее неизвестно будет ли шестнадцатеричная запись числа  $N$  (после отбрасывания незначащих нулей) иметь длину, кратную трём. Если длина не кратна трём выходит, что выделяется префикс из первых значащих шестнадцатеричных цифр (одной или двух), который переводится отдельно, а остальные цифры сгруппируются по тройкам, и каждая такая тройка будет переведена в четвёрку восьмеричных цифр. Поэтому считаем сколько незначащих нулей попало во вводе, вычитаем их количество из  $L$  и переводим, зная кратность количества значащих цифр трём.

### Код возможного решения

```
program DigitQty1011 (input, output);
var
    C : char;
```

```

    D , L, TEMP, ANSWER : word;
function Process3Digit() : word;
var    I : word;
        C : char;
TEMP : word;
(* найти количество вхождений D в переведённые три 16-тиричные цифры без незначащих нулей *)
begin
    TEMP := 0;
    for I := 1 to 3 do begin
        read(C);
        dec(L);
        if ((C <= '9') and (C >= '0')) then TEMP := TEMP * 16 + ord(C) - ord('0')
        else TEMP := TEMP * 16 + ord(C) - ord('A') + 10;
    end; (* for *)
    I := 0;
    if (TEMP mod 8 = D) then inc(I);
if (TEMP div 8 mod 8 = D) then inc(I);
    if (TEMP div 64 mod 8 = D) then inc(I);
    if (TEMP div 512 mod 8 = D) then inc(I);
Process3Digit := I
end;

begin
    readln(C);
    D := ord(C) - ord('0');
    ANSWER := 0;
    TEMP := 0;
    readln(L);
    while ((L > 0) and (TEMP = 0)) do begin
        read(C);
        dec(L);
        if ((C <= '9') and (C >= '0')) then TEMP := ord(C) - ord('0')
        else TEMP := ord(C) - ord('A') + 10;
    end; (* while *)
    case (L mod 3) of
    0: begin
        if (TEMP div 8 mod 8 <> 0) and (TEMP div 8 mod 8 = D) then inc(ANSWER);
        if (TEMP mod 8 = D) then inc(ANSWER)
    end; (* обработан префикс в 1 цифру *)
    1: begin
        read(C);
        dec(L);
        if ((C <= '9') and (C >= '0')) then TEMP := TEMP * 16 + ord(C) - ord('0')
        else TEMP := TEMP * 16 + ord(C) - ord('A') + 10;
        if (TEMP div 64 mod 8 <> 0) and (TEMP div 64 mod 8 = D) then inc(ANSWER);
        if (TEMP div 8 mod 8 = D) then inc(ANSWER);
        if (TEMP mod 8 = D) then inc(ANSWER)
    end; (* обработан префикс в 2 цифры *)
    else begin
        read(C);
        dec(L);
        if ((C <= '9') and (C >= '0')) then TEMP := TEMP * 16 + ord(C) - ord('0')
        else TEMP := TEMP * 16 + ord(C) - ord('A') + 10;
        read(C);
        dec(L);
        if ((C <= '9') and (C >= '0')) then TEMP := TEMP * 16 + ord(C) - ord('0')

```

```
        else TEMP := TEMP * 16 + ord(C) - ord('A') + 10;
        if (TEMP div 512 mod 8 <> 0) and (TEMP div 512 mod 8 = D) then inc(ANSWER);
        if (TEMP div 64 mod 8 = D) then inc(ANSWER);
        if (TEMP div 8 mod 8 = D) then inc(ANSWER);
        if (TEMP mod 8 = D) then inc(ANSWER)
    end; (* обработаны первые 3 цифры *)
end; (* case *)
while (L > 0) do begin
    ANSWER := ANSWER + Process3Digit();
end (* while *);
writeln(ANSWER)
end.
```