Temat C28: Systemy liczbowe

1. Obliczanie wartości liczby w zapisie pozycyjnym

   http://eduinf.waw.pl/inf/alg/006_bin/0002.php

   Zbiór podstawowych cech dowolnego systemu pozycyjnego o podstawie p

   System pozycyjny charakteryzuje liczba zwana podstawą systemu pozycyjnego.

   Do zapisu liczby służą cyfry.

   Cyfr jest zawsze tyle, ile wynosi podstawa systemu: 0,1,2,...,(p-1)

   Cyfry ustawiamy na kolejnych pozycjach.

   Pozycje numerujemy od 0 poczynając od strony prawej zapisu.

   Każda pozycja posiada swoją wagę.

   Waga jest równa podstawie systemu podniesionej do potęgi o wartości numeru pozycji.

   Cyfry określają ile razy waga danej pozycji uczestniczy w wartości liczby

   Wartość liczby obliczamy sumując iloczyny cyfr przez wagi ich pozycji

   Dla podstawy większej niż 10 potrzeba więcej cyfr niż 10 – stosuje się oznaczenia literowe!

Przykład:

wagi	1000 10^3	100 10^2	10 10^1	1 10^0
cyfry	**7**	**5**	**8**	**2**
pozycje	3	2	1	0

Wagi 4 pozycji w różnych systemach liczbowych					
Podstawa p	**Wartości wag pozycji**				
	pozycja 4	pozycja 3	pozycja 2	pozycja 1	pozycja 0
2	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$\mathbf{2^0 = 1}$
3	$3^4 = 81$	$3^3 = 27$	$3^2 = 9$	$3^1 = 3$	$\mathbf{3^0 = 1}$
4	$4^4 = 256$	$4^3 = 64$	$4^2 = 16$	$4^1 = 4$	$\mathbf{4^0 = 1}$
5	$5^4 = 625$	$5^3 = 125$	$5^2 = 25$	$5^1 = 5$	$\mathbf{5^0 = 1}$
6	$6^4 = 1296$	$6^3 = 216$	$6^2 = 36$	$6^1 = 6$	$\mathbf{6^0 = 1}$
7	$7^4 = 2401$	$7^3 = 343$	$7^2 = 49$	$7^1 = 7$	$\mathbf{7^0 = 1}$
8	$8^4 = 4096$	$8^3 = 512$	$8^2 = 64$	$8^1 = 8$	$\mathbf{8^0 = 1}$
9	$9^4 = 6561$	$9^3 = 729$	$9^2 = 81$	$9^1 = 9$	$\mathbf{9^0 = 1}$
10	$10^4 = 10000$	$10^3 = 1000$	$10^2 = 100$	$10^1 = 10$	$\mathbf{10^0 = 1}$

Wartość dziesiętna liczby zapisanej w systemie pozycyjnym o podstawie p za pomocą ciągu cyfr

$C_{n-1}C_{n-2}...C_2C_1C_0$ wynosi $C_{n-1}\,p^{n-1} + C_{n-2}\,p^{n-2} + ... + C_2\,p^2 + C_1\,p^1 + C_0\,p^0$

gdzie:

C - cyfra danego systemu o podstawie p
C_i - cyfra na i-tej pozycji, i = 0,1,2,...,n-1
n - ilość cyfr w zapisie liczby
p - podstawa systemu pozycyjnego

Algorytm obliczania wartości liczby pozycyjnej

Specyfikacja problemu

Dane wejściowe

p - podstawa systemu pozycyjnego zapisu liczby, $p \in N$, $p \in \{2,3,...,10\}$
s - tekst zawierający ciąg znaków ASCII przedstawiających cyfry.

Dane wyjściowe

Liczba L będąca wartością liczby o podstawie *p* i zapisanej w postaci ciągu znaków *s*. $L \in N + \{0\}$

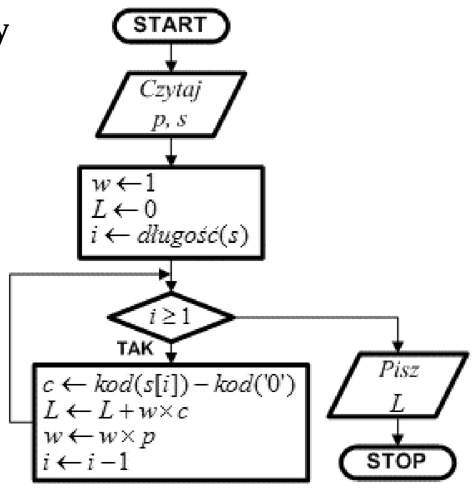
Zmienne pomocnicze i funkcje

w	- wagi kolejnych pozycji, $w \in N$
c	- przechowuje wartość cyfry, $c \in N + \{0\}$
i	- numery pozycji znaków w *s*, $i \in N$
kod(*znak*)	- funkcja zwraca kod ASCII znaku
długość(*tekst*)	- zwraca liczbę znaków zawartych w tekście

Lista kroków

K01:	**Czytaj** *p* i *s*
K02:	$w \leftarrow 1$; $L \leftarrow 0$
K03:	**Dla** *i* = długość(*s*), długość(*s*) - 1,...,1 **wykonuj** K04...K06.
K04:	$c \leftarrow$ kod(*s*[*i*]) - kod('0')
K05:	$L \leftarrow L + w \times c$
K06:	$w \leftarrow w \times p$
K07:	**Pisz** L
K08:	**Zakończ**

Schemat blokowy



2. Przeliczenia na inny zapis pozycyjny

Metoda przeliczania liczb

Problem sprowadza się do znalezienia kolejnych cyfr zapisu liczby w systemie docelowym.

Do wydobycia poszczególnych cyfr C_i , i = 0,1,2,...,n-1, są nam potrzebne dwa działania:

- div - dzielenie całkowitoliczbowe - określa ile całkowitą ilość razy dzielnik mieści się w dzielnej, na przykład:
  9 **div** 4 = 2, gdyż 4 mieści się w 9 2 razy.
- mod - reszta z dzielenia całkowitoliczbowego, na przykład:
  9 **mod** 4 = 1, gdyż 4 mieści się w 9 dwa razy, co daje 8 i pozostaje reszta 1.
  Reszta z dzielenia jest zawsze mniejsza od dzielnika

Praktycznie działania te wykonujemy w słupku dzieląc całkowitoliczbowo liczbę przez podstawę systemu i wypisując reszty z dzielenia. Gdy rachunki zakończymy, otrzymane reszty odczytujemy w kierunku z dołu do góry otrzymując kolejne cyfry zapisu liczby.

Przykład:

Przedstawić w systemie czwórkowym liczbę $2743_{(10)}$.

2743 **div** 4 = 685 i reszta 3
685 **div** 4 = 171 i reszta 1
171 **div** 4 = 42 i reszta 3
42 **div** 4 = 10 i reszta 2
10 **div** 4 = 2 i reszta 2
2 **div** 4 = 0 i reszta 2 - koniec, ponieważ wynik dzielenia wynosi 0

$2743_{(10)} = 222313_{(4)}$.

Przedstawić w systemie dziewiątkowym liczbę $35921_{(10)}$.

35921 **div** 9 = 3991 i reszta 2
3991 **div** 9 = 443 i reszta 4
443 **div** 9 = 49 i reszta 2
49 **div** 9 = 5 i reszta 4
5 **div** 9 = 0 i reszta 5 - koniec

$35921_{(10)} = 54242_{(9)}$.

Przedstawić w systemie trójkowym liczbę $325748_{(10)}$.

325748 **div** 3 = 108582 i reszta 2
108582 **div** 3 = 36194 i reszta 0
36194 **div** 3 = 12064 i reszta 2
12064 **div** 3 = 4021 i reszta 1
4021 **div** 3 = 1340 i reszta 1
1340 **div** 3 = 446 i reszta 2
446 **div** 3 = 148 i reszta 2
148 **div** 3 = 49 i reszta 1
49 **div** 3 = 16 i reszta 1
16 **div** 3 = 5 i reszta 1
5 **div** 3 = 1 i reszta 2
1 **div** 3 = 0 i reszta 1 - koniec

$325748_{(10)} = 121112211202_{(3)}$.

Algorytm przeliczania liczb na inny system pozycyjny

Specyfikacja problemu

Dane wejściowe

L - przeliczana liczba, L ∈ N + {0}

p - podstawa docelowego systemu pozycyjnego, *p* ∈ N, *p* ∈ {2,3,...,10}

Dane wyjściowe

Ciąg znaków s reprezentujący zapis liczby L w systemie pozycyjnym o podstawie *p*.

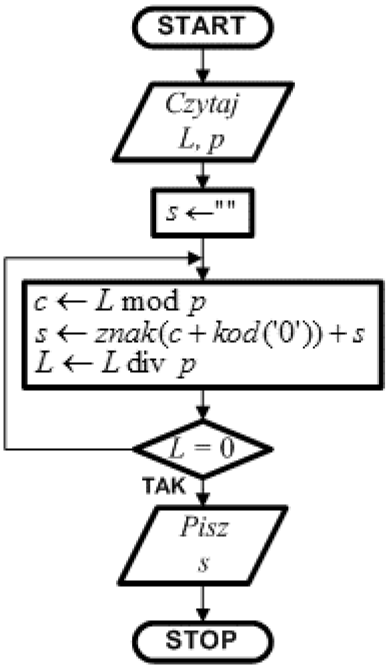
Zmienne pomocnicze i funkcje

s - przechowuje docelowy zapis liczby.

c - przechowuje wartość cyfry, *c* ∈ N + {0}

kod(*znak*) - funkcja zwraca kod ASCII znaku

znak(*kod*) - zwraca znak ASCII o podanym kodzie



K01: **Czytaj** L i *p*

K02: *s* ← ""

K03: *c* ← L **mod** *p*

K04: *s* ← znak(*c* + kod('0')) + *s*

K05: L ← L **div** *p*

K06: **Jeśli** L = 0, **to pisz** *s* i **zakończ**
Inaczej idź do K03.

Odczytujemy liczbę L, którą chcemy przeliczyć oraz podstawę p docelowego systemu pozycyjnego. Podany algorytm pracuje poprawnie tylko dla podstaw p od 2 do 10 (dla większych podstaw należy modyfikować zwracany kod zgodnie z układem liter w kodach ASCII – dodać 7!).

Wyliczone cyfry będziemy odkładać w zmiennej łańcuchowej s. Inicjujemy ją pustym tekstem.

Rozpoczynamy pętlę warunkową, która będzie wykonywana, aż liczba L osiągnie wartość 0.
Wewnątrz pętli obliczamy wartość ostatniej cyfry liczby L i umieszczamy wynik w zmiennej c.

Aby wstawić cyfrę do zmiennej łańcuchowej s musimy ją wyrazić za pomocą kodu ASCII. Dlatego w wyrażeniu wyliczamy kod znaku cyfry jako sumę wartości cyfry oraz kodu cyfry 0. Na przykład dla cyfry 5 otrzymamy kod 5 + 48 = 53 (cyfra 0 ma w ASCII kod 48). Znak o kodzie 53 to właśnie cyfra 5.

Obliczony kod cyfry przekształcamy w znak i łączymy z zawartością łańcucha s.
Bardzo ważna jest tutaj kolejność łączenia. Cyfra musi być dopisana przed poprzednio wyliczonymi cyframi, ponieważ algorytm wyznacza cyfry od końca zapisu liczby.

Po dołączeniu cyfry do łańcucha liczbę L dzielimy całkowitoliczbowo przez p i przechodzimy do sprawdzenia warunku zakończenia pętli. Jeśli po operacji dzielenia liczba L nie jest równa zero, to nie zostały jeszcze wyznaczone wszystkie cyfry, zatem pętla kontynuuje się. Jeśli natomiast liczba L jest równa zero, zmienna s zawiera komplet cyfr liczby w docelowym systemie pozycyjnym. Wychodzimy z pętli, wypisujemy zawartość łańcucha s i kończymy algorytm.

3. Szybka zamiana między systemami o podstawach 2,4,8 i 16
   Skorzystamy z zależności między reprezentacjami liczb w systemie o podstawie 2, podstawie $4 = 2 \cdot 2$ oraz podstawie $8=2 \cdot 2 \cdot 2$ i $16=2 \cdot 2 \cdot 2 \cdot 2$. Skoncentrujmy się najpierw na systemach o podstawach 2 i 4:
   - Reprezentację liczby w systemie czwórkowym można uzyskać z jej reprezentacji w systemie binarnym, (czyli o podstawie 2), wybierając od końca pary cyfr i zamieniając je na ich czwórkowe reprezentacje;
   - Reprezentację liczby w systemie binarnym można uzyskać z jej reprezentacji w systemie czwórkowym, zamieniając każdą cyfrę czwórkową na jej dwucyfrową reprezentację binarną.
   - Ponieważ $8=2^3$, analogiczna własność zachodzi dla konwersji między systemem binarnym a systemem ósemkowym, z tą różnicą, że zamiast bloków 2 cyfr rozważamy bloki o długości 3 cyfr.
   - Dla systemu heksadecymalnego bloki mają długość 4 cyfr.
4. Zadania
   - (8.1) Napisz program realizujący obliczanie wartości dziesiętnej liczby s podanej przez użytkownika o podstawie p podanej przez użytkownika, dla p od 2 do 16.
     W programie utwórz i wykorzystaj funkcję **int wartosc(string s, int p)** zwracającą wartość dziesiętną liczby s o podstawie p
   - (8.2)
   - (8.3) Napisz program realizujący przeliczanie wartości liczby dziesiętnej L podanej przez użytkownika na liczbę o podstawie p podanej przez użytkownika, dla p od 2 do 16.
     W programie utwórz i wykorzystaj funkcję **string nasystem(int L, int p)** zwracającą postać liczby L zapisaną w systemie o podstawie p
   - (8.4)