## Wyprowadzenie schematu Hornera

Obliczenia wartości liczby dokonujemy na podstawie wzoru:

Cn-1Cn-2...C2C1C0 = Cn-1 pn-1 + Cn-2 pn-2 + ... + C2 p2 + C1 p1 + C0 p0

W zastosowaniach informatycznych korzysta się z innego rozwiązania, zwanego schematem Hornera. Właściwie schemat ten ma zastosowanie przy wyznaczaniu wartości wielomianu, lecz jeśli przyjrzymy się dokładnie powyższemu wzorowi, to na pewno zauważymy podobieństwo do wzoru na wartość wielomianu:

W(x) = an-1 xn-1 + an-2 xn-2 + ...+ a2 x2 + a1 x+ a0

Współczynniki ai dla i = 0,1,2,...,n-1 odpowiadają wartościom cyfr C. Natomiast kolejne potęgi zmiennej x to oczywiście potęgi podstawy p. Schemat Hornera wyznaczymy dla 5-cio cyfrowej liczby (dla n-cyfrowej zasada jest identyczna). Liczba zapisana jest w systemie pozycyjnym o podstawie p ciągiem cyfr C4C3C2C1C0 i ma wartość:

L = C4 p4 + C3 p3 + C2 p2 + C1 p1 + C0 p0

Ponieważ p1 = p oraz p0 = 1, powyższy wzór można nieco uprościć i zapisać go w postaci:

L = C4 p4 + C3 p3 + C2 p2 + C1 p + C0

Wyprowadzamy przed nawias wspólny czynnik p:

L = p (C4 p3 + C3 p2 + C2 p + C1) + C0

Zwróć uwagę, iż wyrażenie w nawiasie ma niższy stopień. Znów wyprowadzamy przed nawias wspólny czynnik p.

L = p (p (C4 p2 + C3 p + C2) + C1) + C0

I jeszcze raz:

L = p (p (p (C4 p + C3) + C2) + C1) + C0

I po raz ostatni:

L = p (p (p (p (C4)+ C3) + C2) + C1) + C0

Ze względu na przemienność operacji mnożenia otrzymany wzór możemy zapisać w postaci:

L = ((((C4) p + C3) p + C2) p + C1) p + C0

Teraz wartość liczby obliczamy wyliczając wartości wyrażeń w kolejnych nawiasach:

L0 = C4 - wartość początkowa  
L1 = L0 p + C3 = C4 p + C3  
L2 = L1 p + C2 = (C4 p + C3) p + C2 = C4 p2 + C3 p + C2 L3 = L2 p + C1 = (C4 p2 + C3 p + C2) p + C1 = C4 p3 + C3 p2 + C2 p + C1   
L4 = L3 p + C0 = (C4 p3 + C3 p2 + C2 p + C1) p + C0 = C4 p4 + C3 p3 + C2 p2 + C1 p + C0

Zwróć uwagę na sposób wyliczania wartości liczby. Wyraźnie widoczny jest pewien schemat postępowania. Najpierw za wartość liczby przyjmujemy C4. Następnie do wyczerpania pozostałych cyfr wykonujemy te same obliczenia: nową wartość otrzymujemy mnożąc poprzednią wartość przez podstawę systemu i dodając kolejną cyfrę. Rachunki kończymy po dodaniu ostatniej cyfry zapisu liczby.

Schemat ten nosi nazwę schematu Hornera.

**Przykład:**

Obliczyć za pomocą schematu Hornera wartość liczby piątkowej 4223213(5).

L0 = 4  
L1 = 4 • 5 + 2 = 22  
L2 = 22 • 5 + 2 = 112  
L3 = 112 • 5 + 3 = 563  
L4 = 563 • 5 + 2 = 2817  
L5 = 2817 • 5 + 1 = 14086  
L6 = 14086 • 5 + 3 = **70433** - koniec, ponieważ wyczerpaliśmy wszystkie cyfry

4223213(5) = 70433(10).

## Zalety schematu Hornera

Po co to wszystko jest potrzebne? Po pierwsze oszczędność w mnożeniu. Sprawdźmy. Dla pięciu cyfr musimy wykonać następujące rachunki przy zastosowaniu standardowego wzoru:

L = C4 p p p p + C3 p p p + C2 p p + C1 p + C0

Daje to w sumie 10 mnożeń i 4 dodawania. Ten sam rachunek schematem Hornera prowadzi do wykonania 4 mnożeń i 4 dodawań. Mniej mnożeń oznacza większą efektywność algorytmu Hornera ponieważ mnożenie zajmuje procesorowi komputera więcej czasu od dodawania. Drugą zaletą jest sposób przetwarzania cyfr. Bierzemy je kolejno jedna po drugiej z ciągu wejściowego aż do napotkania końca zapisu. Ponieważ taka kolejność cyfr jest zwykle zgodna z kolejnością ich przechowywania w łańcuchu tekstowym, zatem sposób ten daje nam kolejne przyspieszenie i uproszczenie działania algorytmu (w poprzednim algorytmie cyfry przetwarzaliśmy w kierunku odwrotnym poczynając od ostatniej w zapisie).

## Algorytm

Podsumujmy podane dotychczas informacje w formie algorytmu.

## Specyfikacja problemu

#### Dane wejściowe

|  |  |
| --- | --- |
| *p* | - podstawa systemu pozycyjnego zapisu liczby *p* ∈ N,   *p* ∈ {2,3,...,10} |
| *s* | - tekst zawierający ciąg znaków ASCII przedstawiających poprawny zapis liczby. |

#### Dane wyjściowe

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

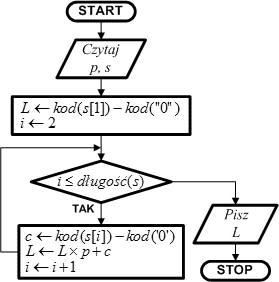
#### Zmienne pomocnicze i funkcje

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

## Lista kroków

|  |  |
| --- | --- |
| K01: | **Czytaj** *p* i *s* |
| K02: | L ← kod(*s*[1]) - kod("0") |
| K03: | **Dla** *i* = 2,3,...,długość(*s*) **wykonuj** K04...K05 |
| K04: | c ← kod(*s*[*i*]) - kod("0") |
| K05: | L ← L × *p* + *c* |
| K06: | **Pisz** L |
| K07: | **Zakończ** |

## Schemat blokowy



Odczytujemy podstawę p systemu liczbowego, w którym zapisana jest liczba. Podstawa musi należeć do zakresu od 2 do 10. Następnie odczytujemy ciąg znaków s reprezentujących cyfry. W zmiennych łańcuchowych pozycje znaków są numerowane od 1 (w C++, Pythonie i JavaScript od 0) począwszy od strony lewej do prawej.

Po odczytaniu danych wejściowych inicjujemy zmienne robocze. Początkowa wartość liczby L ustawiana jest na wartość pierwszej cyfry zapisu. Zmienna i steruje pętlą iteracyjną. Wprowadzamy do niej indeks drugiego znaku odczytanego łańcucha s i rozpoczynamy pętlę.

Pętla wykonuje się o jeden raz mniej niż liczba znaków w łańcuchu s. W pętli wyznaczamy wartość kolejnej cyfry w c.  Za nową wartość liczby L przyjmujemy poprzednią wartość pomnożoną przez p i zwiększoną o wartość cyfry c (schemat Hornera). Zwiększamy indeks i przechodzimy na początek pętli.

Po zakończeniu pętli w zmiennej L mamy wartość liczby. Wypisujemy ją i kończymy algorytm.