

lab_4 – Wprowadzenie

Problem: prezentacja liczb w postaci znakowej dziesiętnej

liczba 123 => '123' dziesiętnie

Algorytm:

Wielokrotne dzielenie przez podstawę systemu (10) i zapis reszt z dzielenia w postaci znakowej w odpowiedniej kolejności

Przykład: liczba 357

<u>Krok</u>	<u>Dzielną</u>	<u>Iloraz</u>	<u>Reszta</u>	<u>Znak</u>
1	357	35	7	'7'
2	35	3	5	'5'
3	3	0	3	'3'

Warunkiem stopu w algorytmie jest wyzerowanie się ilorazu

Zamiana reszty na cyfrę dziesiętną (znak):

aaaa – liczba [0..9]

0	=>	'0'	kod ASCII dec 48, hex 0x30
1	=>	'1'	kod ASCII dec 49, hex 0x31
		...	
9	=>	'9'	kod ASCII dec 57, hex 0x39

code = aaaa + 48 (0x30, '0')

Zapis liczby w postaci łańcucha znaków:

- utworzenie napisu (ciągu znaków) poprzez zapisanie uzyskanych cyfr w odpowiedniej kolejności
- problem: zmienna liczba znaków
- problem: znaki pojawiają się w kolejności od najmniej znaczącej cyfry do najbardziej znaczącej ('7', '5', '3') a mają utworzyć napis '357' – trzeba je „odwrócić” (np. poprzez użycie stosu lub danej o dostępie swobodnym (tablicy))

Zapis od prawej do lewej:

7	7
18	18
246	246
3579	3579

efekt: łatwiejsze porównanie liczb i oszacowanie wielkości

Realizacja:

- pętla **do..while()**
- dzielenie całkowite **DIV %ebx**:
 - dzielnik **ebx** (32 bity)
 - reszta **edx** (32 bity)
 - iloraz **eax** (32 bity)
 - dzielna **edx:eax** (**64 bity!**)
- problem: jak zwrócić rezultat konwersji? – funkcja dostaje jako argument adres miejsca w którym ma zapisać znaki i zapisuje je w kolejności od prawej do lewej (funkcja dostaje adres miejsca przeznaczonego na najmniej znaczącą cyfrę dziesiętną i sama zmniejsza ten adres przy zapisywaniu kolejnych znaków)