Міністерство освіти і науки України

Донецький національний університет імені Василя Стуса

Фізико-технічний факультет

Кафедра «Комп’ютерні технології»

**НАВЧАЛЬНА ПРАКТИКА**

**ЗВІТ**

виконавець:

студент 1 курсу групи КН-Б кафедри КТ

Суханов Андрій Олександрович

керівник практики:

доцент кафедри КТ

Парамонов А.І.

Вінниця – 2020

ЗМІСТ

[**ВСТУП** 3](#_Toc31313948)

[**Огляд арифметичних дій над числами з плаваючою точкою** 4](#_Toc31313949)

[**Прямий, обернений та додатковий коди у комп’ютері** 5](#_Toc31313950)

[**МАНТИСА ТА ПОРЯДОК** 6](#_Toc31313951)

[**Множення та ділення двійкових чисел із фіксованою комою** 7](#_Toc31313952)

[**Подання чисел у форматі з фіксованою комою** 8](#_Toc31313953)

[**ЗАСТОСУВАННЯ ТА РЕАЛІЗАЦІЯ ЧИСЕЛ З ФІКСОВАНОЮ КРАПКОЮ** 9](#_Toc31313954)

[**ВИСНОВКИ** 10](#_Toc31313956)

[**Список використаних джерел** 11](#_Toc31313957)

[**Додатки** 12](#_Toc31313958)

**ВСТУП**

Програмування. Чи не найактуальніша професія сьогодення?

Спеціалізація у цій сфері вимагає знання іноземних мов, навичок та вмінь у мовах програмування. Стрімкий розвиток технологій у світі потребує сильних фахівців у цій галузі.

На мою думку, логічне мислення та всебічна розвиненість людини -

ось найголовніші фактори для вправного орієнтування у цій діяльності. Системні програмісти, прикладні програмісти та веб-програмісти – усе це види спеціалістів, що називають себе «айтішніками».

Чому саме цей шлях я обрав? По-перше, програмування в Україні

швидко розвивається, а саме тому на ринку праці не вистачає досвідчених та кваліфікованих фахівців. Отже, така професія, як мінімум, забезпечить високу зарплатню та шалений кар’єрний ріст. По-друге, стати програмістом – це моя мрія з дитинства. Будучи ще малечею, я завжди проявляв зацікавленість до комп’ютера, його внутрішніх програм та деталей. Можливо, у найближчому майбутньому я розроблю програму, що допоможе людству вирішити глобальні проблеми, але поки це лише моя мрія. Чому б не спробувати? По-третє, знаючі IT-спеціалісти завжди працюють у популярних компаніях. Веб-програмісти розробляють сайти для всесвітньо відомих брендів, магазинів, корпорацій та спілок. Таким чином, вони набувають досвіду та отримують чималу зарплатню. Хіба це не добре? Забезпечити усім необхідним себе, свою сім’ю, родину та жити у достатку? Адже життя спішить та потрібно вміти ним насолоджуватися…

Мені подобається уривок вірша Ліни Костенко «Життя іде і все без

коректур». Вона зазначає: «Єдиний, хто не втомлюється, – час. А ми живі, нам треба поспішати». І справді, життя – це те, що ми робимо, це те, де ми живемо. Потрібно не зволікати, поспішати знайти себе та займатися улюбленим ділом…

Отже, професію програміста я обрав недарма: мій вибір заснований не

на якихось дрібницях, а на реальній зацікавленості у цій діяльності. Я маю дуже велике бажання розвиватися у програмуванні, адже це дуже модно і перспективно у наші дні. Але найголовніше, ­­­– ця професія повинна бути до душі, адже кожна робота – це частина нашого життя і не потрібно витрачати час на те, від чого не отримуєш задоволення та на те, що нецікаве.

**Огляд арифметичних дій над числами з плаваючою точкою**

У десятковій системі існує багато форм подання чисел. Ми,

користуючись цим, самі вибираємо ту або іншу форму. Наприклад, число 0,25 можна відобразити як 1/4 при виконанні операцій з такими ж дробовими числами. Можна відобразити як 25\*10-2  або 2500\*10-4 , або 0,0025\*102.

Усю різноманітність запису чисел поділяють на природну і

нормалізовану (нормальну) форми.

При природній формі число записують у природньому вигляді,

наприклад, 125 – ціле число, а 0,125 – правильний дріб, 125,125 – неправильний дріб. При нормалізованій формі запис числа може примати різний вигляд і його форма залежить від прийнятих чинних правил(обмежень). Наприклад, 12500 може бути записано тільки як 12,5\*103 або 0,125\*105 і т.д.

Така велика кількість форм подання чисел може стати причиною

ускладнення пристроїв і алгоритмів функціювання цифрових автоматів.

Щоб уникнути цього, у цифрових автоматах прийнято свої

нормалізовані форми запису і відображення чисел, що дозволяє мінімізувати вартість апаратних і алгоритмічних засобів.

Найбільше поширення отримали формати з фіксованою крапкою та

рухомою комою (крапкою).

В [інформатиці](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), число з нерухомою — це представлення дійсного числа,

що має фіксовану кількість чисел після (іноді перед) відокремлювальної коми. Представлення з нерухомою комою можна порівняти зі складнішим (і вимогливішим до обчислень) представлень з рухомою комою.

Числа з нерухомою комою корисні для подання дробових чисел,

зазвичай з основою 2 або 10, коли [центральний процесор](https://uk.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) не має [математичного співпроцесора](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B8%D0%B9_%D1%81%D0%BF%D1%96%D0%B2%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) або нерухома кома має покращену швидкодію або точність. Найдешевші [вбудовані](https://uk.wikipedia.org/wiki/%D0%92%D0%B1%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [мікропроцесори](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) та [мікроконтролери](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80) не мають співпроцесора.

**Існують такі операції над числами з фіксованою крапкою:**

• Додавання і віднімання чисел з нерухомою комою — це звичайні додавання і віднімання.

• Аналогічно з множенням і діленням на цілочисельну константу.

• Множення і ділення відрізняються від цілочисельних на константу.

• Для інших операцій, крім звичайних [рядів Тейлора](https://uk.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0) і ітераційних методів, широко застосовуються обчислення за таблицею.

Детальніше ці операції ми розглянемо далі.

**Прямий, обернений та додатковий коди у комп’ютері**

У сучасних комп’ютерах здебільшого застосовуються два формати

зображення чисел: числа з фіксованою комою (fixed point) і плаваючою комою (floating point).

Числа з фіксованою комою – це формат з незмінним розташуванням

коми, що відокремлює цілу частину числа від дробової.

n біт r біт

|  |  |  |
| --- | --- | --- |
| Біт знака | Ціла частина | Дробова частина |

**Рис.1.1**

Самий лівий розряд чисел називається знаковим. Знаковий розряд

дорівнює 0 для додатних чисел, та 1 – для від’ємних чисел. В числах з фіксованою комою положення коми не міняється при проведенні арифметичних операцій. Для спрощення операцій над такими числами положення коми зазвичай фіксується або перед старшим цифровим розрядом (для дробових чисел), або після молодшого розряду (для цілих чисел).

Для зображення додатних та від’ємних чисел в комп’ютері

застосовується прямий, обернений та додатковий коди. Ідея полягає в тому, щоб замінити операцію віднімання операцією додавання.

В прямому коді додатні і від’ємні числа відрізняються тільки знаковим

розрядом, наприклад:

+5→ 0. 101

-5→ 1. 101

+127→ 0. 1111111

-127→ 1. 1111111

Прямий код використовується для введення/виведення інформації, а

також для збереження коду в пам’яті.

Обернений код застосовується для зображення від’ємних чисел заміною

усіх значущих розрядів на протилежні:

+5→ 0. 101

-5→ 1. 010

+127→ 0. 1111111

-127→ 1. 0000000

Ця операція еквівалентна відніманню цього числа від 2n -1. ­Якщо

обернений код використовується в арифметичних операціях додавання та віднімання, то результат також будемо отримувати в оберненому коді (при наявності циклічного переносу від знакового розряду в молодший розряд).

Недоліком оберненого коду є необхідність використання циклічного переносу, який потребує додаткового витрату часу. Цього недоліку позбавлений додатковий код.

Додатковий код можна отримати додаванням одиниці до оберненого

коду, або відніманням числа від 2n.

+5→ 0. 101

-5→ 1. 011

+127→ 0. 1111111

-127→ 1. 0000001

Сенс введення формату додаткового коду ( як і оберненого) в тім, що в

цьому форматі операція віднімання замінюється більш простою з точки зору реалізації операцією додавання.

**МАНТИСА ТА ПОРЯДОК**

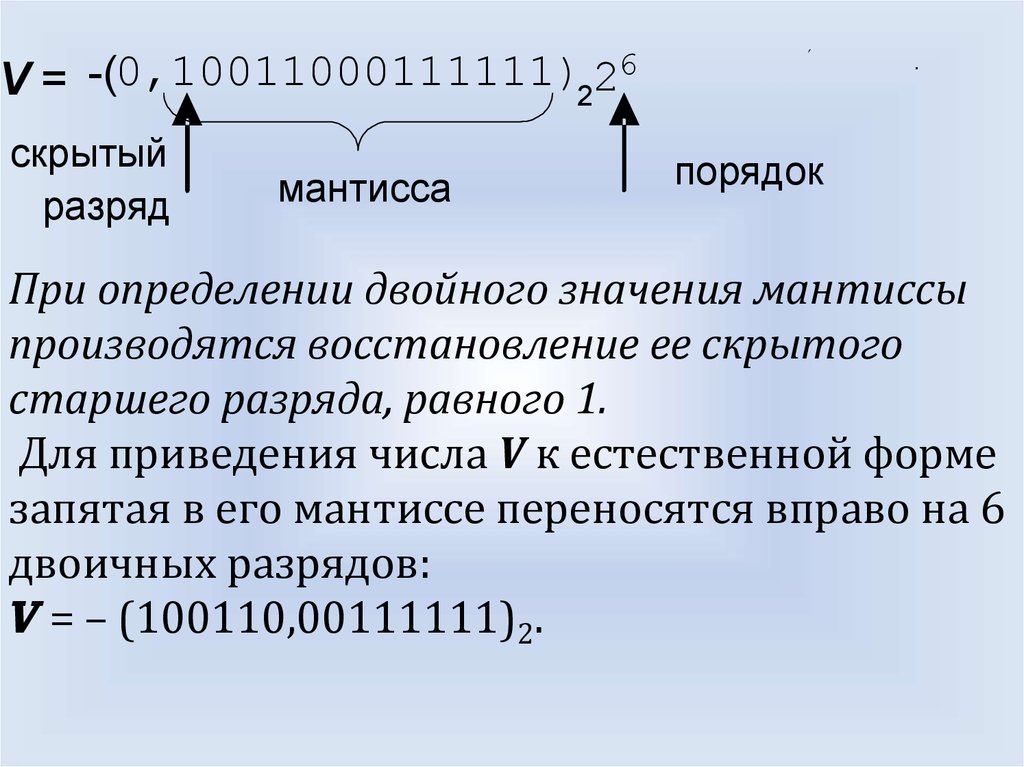
8 біт 23 біта

|  |  |  |
| --- | --- | --- |
| Біт знака | порядок | мантиса |

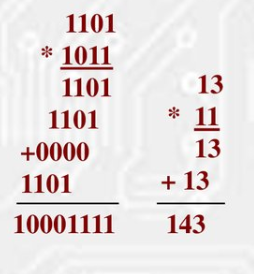
**Рис.1.2**

В прикладних задачах досить часто доводиться оперувати з дуже великим діапазоном чисел (наприклад, від 10-20 до 10+20). Записати в пам’яті подібні числа, враховуючи всі значущі цифри, неможливо. В цьому разі використовується формат з плаваючою комою, коли число розбивається на дві групи цифр. Перша група називається **мантисою** (М), друга – **порядком** (Р). Тоді число можна представити у вигляді:



Отже, як підсумок, можна зазначити: при операціях додавання і віднімання для чисел з плаваючою комою потрібно спочатку вирівняти порядки і тільки потім проводити відповідну операцію з мантисами. А при операціях множення чи ділення, ці операції проводяться над мантисами, а порядки алгебраїчно додаються або віднімаються. Арифметичні операції в двійковій системі суттєво спрощені відносно десяткової системи.  **Рис.1.3** (Обрахування мантиси)

**Множення та ділення двійкових чисел із фіксованою комою**

***Множення*** двійкових чисел виконують у

прямому коді (див. вище). Знак добутку визначають за знаковими розрядами множників згідно з загальновідомим правилом: якщо знаки операндів однакові, то результат додатний, у протилежному випадку – знак добутку від’ємний. Цей алгоритм реалізується за допомогою суми за модулем 2. Знак добутку двох чисел не впливає на алгоритм виконання операції множення модулів цих чисел.

**Рис. 2.1**

Часто використовують спосіб множення, процедура якого аналогічна

процедурам множення вручну. У цьому випадку результат отримують додаванням часткових добутків. Кожен частковий добуток удвічі перевищує попередній, що відповідає його зсуванню ліворуч на один розряд.

Характерно, що у загальному випадку розрядність добутку двійкових

чисел удвічі перевищує розрядність співмножників. Якщо у множенні беруть участь мантиси, тобто правильні дроби, то молодші розряди, що виходять за межі розрядної сітки, можуть бути відкинуті без округлення або з округленням.

***Ділення*** також виконується способом, аналогічним застосовуваним

діленням вручну, що ілюструє приклад ділення двох чисел:

506 : 23 = 22, тобто:

0.111111010 : 0.10111 = 0.10110.

Знак частки визначають аналогічно до знаку добутку. Застосоване при діленні віднімання дільника виконують шляхом додавання його додаткового коду.



**Рис.2.2**

**Подання чисел у форматі з фіксованою комою**

Зображення комп’ютерного числа − це подання числа N у вигляді n-

розрядного коду у наперед заданому форматі за встановленими правилами.

При поданні числа у форматі з фіксованою комою весь формат n-розрядної сітки заздалегідь розбивається на три частини:

- два лівих розряди призначені для подання знаку числа:

1) 00 – додатне число;

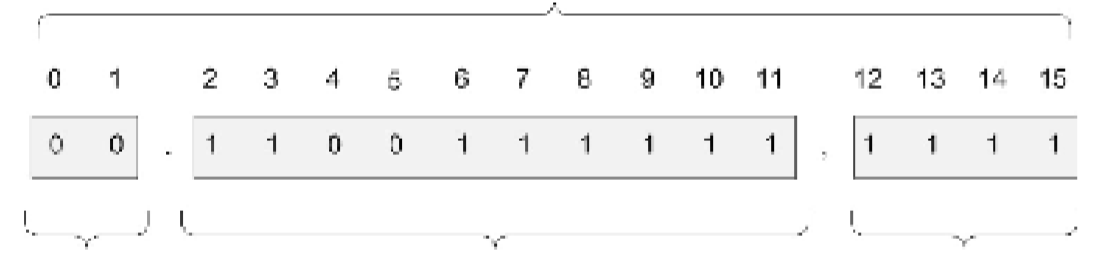
2) 11 – від’ємне число;

- наступні k розрядів використовуються для розміщення цілої

частини числа;

- наступні n−(k+2) розрядів призначені для розміщення дробової

частини числа.

 n-розрядів

Знак k-розрядів(ціла частина) n-(k-2)розрядів

(дробова частина)

**Рис.3.1** – Формат числа з фікс. крапкою

Дійсно, положення коми строго фіксовано в розрядній сітці автомата.

При запису чисел у форматі з фіксованою крапкою, якщо число виходить за межі розрядної сітки числа, молодші розряди відкидаються. При цьому, можуть виникати похибки, зокрема, ділення на 0, що приводить до невизначеності і необхідності втручатися у розрахунки. Однак, при роботі ПК у форматі з фіксованою крапкою швидкість виконання арифметичних і логічних операцій зростає. Похибка подання чисел зменшується при правильному виборі масштабних коефіцієнтів.

**ЗАСТОСУВАННЯ ТА РЕАЛІЗАЦІЯ ЧИСЕЛ З**

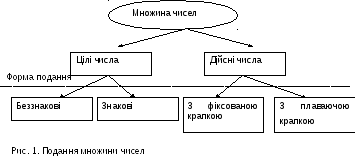
**ФІКСОВАНОЮ КРАПКОЮ**

* Щоб забезпечити мінімальну підтримку дробових чисел на процесорі, що підтримує лише цілі числа (наприклад, на [мікроконтролері](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80)). Якщо не вирішуються [некоректні задачі](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%B5%D0%BA%D1%82%D0%BD%D0%BE_%D0%BF%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B0), часто можна обійтися нерухомою комою — важливо тільки підібрати відповідну ціну (вагу) молодшого розряду для кожної з величин.
* Для прискорення обчислень в місцях, де не потрібна висока точність. У більшості сучаснихпроцесорів нерухома кома апаратно не реалізована, але навіть програмна нерухома кома дуже швидка.
* Для запису чисел, які за своєю природою мають постійну [абсолютну похибку](https://uk.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D0%BE%D0%BB%D1%8E%D1%82%D0%BD%D0%B0_%D0%BF%D0%BE%D1%85%D0%B8%D0%B1%D0%BA%D0%B0): координати в програмах верстки, [позначки часу](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B7%D0%BD%D0%B0%D1%87%D0%BA%D0%B0_%D1%87%D0%B0%D1%81%D1%83), грошові суми. Наприклад, здачу в супермаркеті, податки в країні обчислюють з точністю до копійки. А файли метрики шрифти [TeX](https://uk.wikipedia.org/wiki/TeX) використовують 32-бітний рядковий тип з фіксованою комою (12,20). На подібні величини можна віддати і [рухому кому](https://uk.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE_%D0%B7_%D1%80%D1%83%D1%85%D0%BE%D0%BC%D0%BE%D1%8E_%D0%BA%D0%BE%D0%BC%D0%BE%D1%8E) з достатньою кількістю знаків мантиси — але тоді поле порядку стає зайвим.
* Крім того, кома фіксована веде себе абсолютно передбачувано — при підрахунку грошей це дозволяє налагодити різні види [округлення](https://uk.wikipedia.org/wiki/%D0%9E%D0%BA%D1%80%D1%83%D0%B3%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F), а в іграх — найпростіший спосіб реалізувати [мультиплеєр](https://uk.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%BB%D0%B5%D1%94%D1%80) і запис повторів.

Недолік нерухомої коми — дуже вузький діапазон чисел,з

загрозою [переповнення](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D0%BF%D0%BE%D0%B2%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F) на одному кінці діапазону та [втрати точності](https://uk.wikipedia.org/w/index.php?title=%D0%92%D1%82%D1%80%D0%B0%D1%82%D0%B0_%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%96&action=edit&redlink=1) обчислень на іншому. Ця проблема призвела до винаходу рухомої коми. Наприклад: якщо потрібна точність у 3 значущих цифрах, 4-байтова кома фіксована дає діапазон в 6 порядків (тобто, різниця приблизно 106 між найбільшим і найменшим числом), 4-байтове [число одинарної точності](https://uk.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE_%D0%BE%D0%B4%D0%B8%D0%BD%D0%B0%D1%80%D0%BD%D0%BE%D1%97_%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%96) — в 70 порядків.

**Реалізація**. Деякі мови програмування надають вбудовану підтримку чисел з фіксованою комою, оскільки для більшості застосувань двійкове або десяткове представлення чисел з рухомою комою простіше і досить точно. Числа з рухомою комою простіше використовувати через їх більший динамічний діапазон, для них не потрібно попередньо задавати кількість цифр після коми. Якщо ж буде потрібно арифметиці з фіксованою комою, вона може бути реалізована програмістом навіть на мовах типу C і C++, які зазвичай не включають в себе такої арифметики.

****

**ВИСНОВКИ**

* Операція віднімання в обчислювальних пристроях завжди замінюється

операцією додавання з числом протилежного знака (доповненням модуля від’ємника).

* Операції множення і ділення в обчислювальних пристроях

виконуються у вигляді послідовності операцій додавання і зсування.

* Таким чином, виконання усіх арифметичних операцій у ЕОМтак чи

інакше зводиться до виконання послідовності дій, серед яких основною є додавання чисел, допоміжними – зсування, інверсія та ін. Тому всі ЦА, що виконують арифметичні операції, містять у своєму складі один або декілька суматорів.

* Арифметичні операції з рухомою комою, порівняно з операціями

фіксованою комою, складніше, тому виконуються значно повільніше.

* Цифровий автомат, як ми вияснили, завжди спричиняє появу похибок в

розрахунках, величина яких залежить від обмежень, що накладаються на автомат за розрядною сіткою, форм і подання чисел.

* Сукупність двійкових розрядів, призначених для зберігання і обробки

чисел, являє розрядну сітку машини. Розрядна сітка машини визначає формати чисел, якими можна оперувати при обробці інформації. У машині не може бути представлено число, що містить більшу кількість двійкових розрядів, ніж їх є в розрядної сітці машини.

* У цифрових ЕОМ використовуються дві форми подання чисел:

природна (подання чисел з фіксованою комою) і нормальна (подання чисел з плаваючою комою).

* Для переведення чисел із формату з нерухомою комою у звичайний

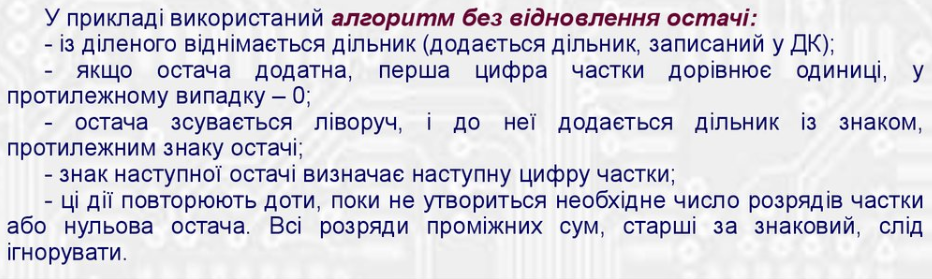
формат і навпаки застосовуються звичайні правила переведення десяткових чисел з однієї позиційної [системи числення](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F) в іншу.

**Список використаних джерел:**

* <http://www.wikipage.com.ua/>
* <https://ppt-online.org/221083>
* <https://studfile.net/preview/3907471/>
* <http://eztuir.ztu.edu.ua/bitstream/handle/123456789/7484/%D0%9F%D0%A2%D0%A6%D0%90.pdf?sequence=1&isAllowed=y>
* <http://eir.zntu.edu.ua/bitstream/123456789/58/1/M03406.pdf>
* <https://studopedia.su/11_114765_chisla-z-fiksovanoyu-komoyu.html>

**Додатки**

ДОДАТОК А. Додаткове пояснення прикладу



**Рисунок А.1 – Ділення без остачі**