МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ВИКОНАННЯ РОЗРАХУНКОВО-ГРАФІЧНОЇ РОБОТИ З ДИСЦИПЛІНИ «ОСНОВИ АЛГОРИТМІЗАЦІЇ» ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 122 КОМП'ЮТЕРНІ НАУКИ

Покровськ - 2018

Методичні вказівки щодо виконання розрахунково-графічної роботи з дисципліни «Основи алгоритмізації» для студентів спеціальності 122 Комп'ютерні науки, денної форми навчання). Патрушев В.О., Патрушева О.І. - Покровськ: ДонНТУ, 2017 рік. – 23 с.

Методичні рекомендації містять загальні положення, мету, завдання, вимоги до структури, змісту, до порядку оформлення розрахунково-графічної роботи по дисципліні «Основи алгоритмізації», що призначені для надання допомоги студентам при виконанні розрахунково-графичної роботи по вказаній дисципліні. Рекомендовано для студентів освітнього ступеню бакалавр» зі спеціальністю 122 «Комп'ютерні науки»

Укладачі: Патрушев В.О., ст.викл.кафедри КН, Патрушева О.І, ас.кафедри КН.

Рецензенти: С.О.Ковальов, декан факультету КНТ, к.т.н., доцент.

Відповідальний за випуск: Є.Є.Федоров., завідувач кафедри КН.

Розглянуто на засіданні кафедри комп'ютерних наук,
протокол № 5 від 27.11.2017 р.

Затверджено навчально-методичним відділом ДонНТУ

ДВНЗ «ДонНТУ», 2018

ЗМІСТ

Вступ

Методичні рекомендації містять вказівки по вивченню окремих тем та завдання до розрахунково-графічної роботи з курсу «Основи алгоритмізації». Матеріал може бути використано при поясненні нового матеріалу, для організації самостійної роботи студентів, для проведення контрольних робіт. Задачі, що надані в посібнику, орієнтовані на послідовне вивчення позиційних систем числення, правил перекладу з однієї системи числення в іншу, виконання основних арифметичних операцій над двійковими числами, вивченню поняття прямий, допоміжний та зворотний коди на прикладі вирішення простих задач.

Розрахунково-графічна робота має тему, мету, загальне завдання до роботи, індивідуальні варіанти завдань, вимоги до оформлення змісту звіту, приклади притаманні визначеній темі та інше.

Головна увага приділяється практичному застосуванню отриманих теоретичних знань за дисципліною «Основи алгоритмізації», а саме: перекладу між системами числення, виконанню складання та віднімання в двійковій системі числення, алгоритмам множення та ділення над двійковими числами.

Підчас виконання розрахунково-графічної роботи студент повинен:

– вивчити відповідні розділи теоретичного лекційного курсу;
– виконати завдання у відповідності до варіанту;
– скласти звіт виконання розрахунково-графічної роботи;
– подати звіт на перевірку викладачу.

Правильність роботи визначає викладач підчас захисту розрахунково-графічної роботи.

1 Мета та завдання розрахунково-графічної роботи студентів

Метою розрахунково-графічної роботи по дисципліні «Основи алгоритмізації» для студентів спеціальності 122 Комп’ютерні науки, згідно вимог освітньо-професійної програми підготовки бакалаврів галузі знань 12 «Інформаційні технології» є:

– закріплення та поглиблення теоретичних знань студентами, що були отримані на лекційних заняття по вказаній дисципліні;

– набуття та вдосконалення студентами практичних навичок у розв’язанні задач, що є характерними для обробки систем числення та типових операцій над ними;

– знайомство з принципами та прийомами роботи з системами числення;

– закріплення, систематизація та поглиблення знань з подальшим застосуванням отриманих знань та вмінь.

Основні завдання лабораторних робіт студентів:

– вивчення теоретичних основ роботи з позиційними системами числення;

– визначення основних методів, прийомів та методологій, що необхідно застосувати для вирішення поставленої задачі;

– розв’язання завдань з застосуванням визначених алгоритмів.

2 Частина 1. Позиційні системи обчислення. Двійкова система обчислення

Мета: Ознайомитися з правилами переведення чисел з однієї системи обчислення в іншу, правилами й особливостями виконання арифметичних операцій у двійковій системі обчислення.

2.1 Загальні положення

Під системою обчислення розуміють сукупність прийомів і правил для запису чисел за допомогою деякого алфавіту символів, які називають цифрами. Загальноприйнятим зараз є позиційне обчислення, у якому значення будь-якої цифри визначається не тільки прийнятою конфігурацією її символу, але і місцем розташування (позицією) у числі. У позиційній системі числення число представляється у виді:

$$A(q) = \sum_{i=-m}^{n-1} a_{ji} q^i , \qquad (1)$$

де a_j – цифра з алфавіту цифр, q – база чи корінь системи, i – номер (вага) позиції (розряду) n – кількість розрядів цілої частини числа, m – кількість розрядів дробової частини числа. У позиційних системах кількість цифр алфавіту дорівнює підставі системи. Вираження (1) для цілих чисел можна представити у виді, відомому за назвою формули Горнера:

$$A(q) = a_{j(n-1)} q + a_{j(n-2)} q + ... + a_0 \qquad (2)$$

Для дробових чисел формула Горнера має вид:

$$A(q) = q^{-1}(a_{j(-1)} + q^{-1}(a_{j(-2)} + ... + q^{-1}(a_{j(-(n-1))} + q^{-1} a_{j(-n)})...)) \quad (3)$$

Якщо $q = 2$, то така система називається двійковою і має у своєму алфавіті відповідно дві цифри: 0, 1.

Якщо $q = 10$, то це усім відома десяткова система, в алфавіті якої використовуються цифри *0, 1, 2, 3, 4, 5, 6, 7, 8, 9*. В обчислювальній техніці для зручності представлення чисел також використовується вісімкова система обчислення ($q = 8$; $a_i = 0, 1, 2, 3, 4, 5, 6, 7$), і шістнадцятикова ($q = 16$; $a_j = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$). Для того, щоб розрізняти в якій системі записане число (однаковому написанню числа в різних системах обчислення відповідають різні числа), у мові програмування асемблер після числа вказуються символи, що визначають систему числення:

- десяткова – D чи ніякого символу;
- двійкова – B (приклад: 01101100B);
- вісімкова – Q чи O (приклад: 45703Q);
- шістнадцятикова – H (приклад: 0F150H).

При записі шістнадцятикового числа, якщо воно починається з цифри більшої 9, то перед ним ставиться нуль, щоб розрізняти число від слова. Щоб відрізняти букву 'O' від нуля останній перекреслюють ∅.

При переведенні чисел також буває корисна таблиця 1.

Методи переведення з однієї системи обчислення в іншу розділяються на дві групи:

а) на основі формули (1); б) на основі формул Горнера (2),(3);

Для переведення за першим методом необхідно цифри і базу системи обчислення, з якої переводиться число, представити в новій системі обчислення і виконати обчислення згідно (1) у новій системі.

Приклад 1. Перевести 123 з десяткової системи в двійкову і навпаки.

$123 = (1\times100) + (2\times10) + (3\times1) = (1\times1100100) +(10\times1010) + (11\times1) = 1111011B$

$1111011B = (1\times64) + (1\times32) + (1\times16) + (1\times8) + (0\times4) + (1\times2) + (1\times1) = 123$

Таблиця 1. Зображення чисел у різних системах обчислення

10-на	2-на	8-на	16-на	2^n	2^{-n}
0	0000	00	0	1	
1	0001	01	1	2	0,5
2	0010	02	2	4	0,25
3	0011	03	3	8	0,125
4	0100	04	4	16	0,0625
5	0101	05	5	32	0,03125
6	0110	06	6	64	0,015625
7	0111	07	7	128	0,0078125
8	1000	10	8	256	0,00390625
9	1001	11	9	512	0,001953125
10	1010	12	A	1024	
11	1011	13	B	2048	
12	1100	14	C	4096	
13	1101	15	D	8192	
14	1110	16	E	16384	
15	1111	17	F	32768	
16	10000	20	10	65536	
17	10001	21	11	131072	

Приклад 2. Перевести дробове число з двійкової системи в десяткову:

$0,101B = (1\times0,5) + (0\times0,25)\cdot + (1\times0,125) = 0,625$

За другим методом перевід цілих і дробових чисел необхідно здійснювати окремо. При переводі цілих чисел з системи числення з базою q_1 в систему числення з базою q_2 необхідно:

- розділити число $A(q_1)$ на q_2; залишок від ділення є нижньою цифрою числа $A(q_2)$;

- розділити цілу частку, отриману при попереднім діленні, на q_2; залишок від цього ділення є черговою цифрою числа $A(q_2)$;

- повторювати операцію ділення доти, поки частка не стане дорівнювати нулю; залишок від кожного ділення є наступною по вазі цифрою числа $A(q_2)$.

Приклад 3. Перевести 123 з десяткової системи обчислення в двійкову.

123:2 = 61 залишок 1 (молодша цифра)
61:2 = 30 залишок 1 (наступна цифра)
30:2 = 15 залишок 0
15:2 = 7 залишок 1
7:2 = 3 залишок 1
3:2 = 1 залишок 1
1:2 = 0 залишок 1 (верхня цифра)
123 = 1111011B

Приклад 4. Перевести 123 з десяткової системи в шістнадцятикову.

123:16 = 7 залишок 11 (чи цифра B) (нижня цифра)
7:16 = 0 залишок 7 (верхня цифра)
123 = 7BH

При переведенні дробових чисел з системи числення з підставою q_1; в систему числення з базою q_2; необхідно:

- помножити число $A(q_1)$ на q_2; ціла частина отриманого результату є верхньою цифрою числа $A(q_2)$;

- помножити дробову частину попереднього результату на q_2; ціла частина отриманого результату є черговою цифрою числа $A(q_2)$;

- повторювати останню операцію доти, поки не буде отримана достатня кількість цифр числа $A(q_2)$, або дробова частина не стане дорівнювати нулю.

Приклад 5. Перевести 0,541 з десяткової системи в двійкову

0,541×2 = 1,082 (верхня цифра перша після коми 1)
0,082×2 = 0,164 (наступна цифра 0)
0,164×2 = 0,328 (наступна цифра 0)
0,328×2 = 0,656 (наступна цифра 0)
0,656×2 = 1,312 (наступна цифра 1)

$0{,}312 \times 2 = 0{,}624$ (наступна цифра 0)

$0{,}624 \times 2 = 1{,}248$ (наступна цифра 1)

0,541 = 0,1000101B

Приклад 6. Перевести 0,541 з десяткової системи в шістнадцятикову.

$0{,}541 \times 16 = 8{,}656$ (верхня цифра 8)

$0{,}656 \times 16 = 10{,}496$ (наступна цифра А)

0,541 = 0,8AH

Правила переведення з двійкової системи обчислення у вісімкову і шістнадцятикову винятково прості, оскільки бази вісімкової і шістнадцятикової систем є цілі ступені двійки. Для перекладу вісімкового числа в двійкову форму досить замінити кожну цифру вісімкового числа відповідним трирозрядним двійковим числом (див. таблицю 1). У такий же спосіб кожна шістнадцятикова цифра замінюється чотирирозрядним двійковим числом.

Приклад 7. Перевести число 753,62Q у двійкову систему.

753,62Q = 111 101 011, 110 010B

Приклад 8. Перевести число 0A6F,1BH у двійкову систему.

0A6F,1BH = 1010 0110 1111, 0001 1011B

Перехід від двійкової до вісімкової (чи шістнадцятикової) системи здійснюють у такий спосіб: рухаючись від коми ліворуч чи праворуч, розбивають двійкове число на групи по три (чотири) розряди, доповнюючи при необхідності нулями крайню ліву групу (для цілої частини числа) і крайню праву групи (для дробової частини числа). Потім кожну групу замінюють відповідною вісімковою (шістнадцятиковою) цифрою з таблиці 1.

Приклад 9. Перевести 101100010,110010B у шістнадцятикову систему.

101100010,110010 = 0001 0110 0010, 1100 1000 = 162,C8H

Для прискорення переведення великих чисел з десяткової системи в двійкову рекомендується спочатку перевести число в шістнадцятикову систему за формулами (2), (3), а потім у двійкову. При цьому істотно скорочується кількість операцій ділення.

2.2 Виконання арифметичних операцій у двійковій системі

Правила виконання арифметичних операцій над двійковими числами задаються таблицями двійкового складання (табл.2), віднімання (табл.3) і множення (табл.4) для однойменних вагових розрядів чисел:

Таблиця 2 $s_i = a_i + b_i + p_i$

a_i	b_i	p_i	s_i	p_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Таблиця 3 $s_i = a_i – b_i - z_i$

a_i	b_i	z_i	s_i	z_{i+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Таблиця 4 $s_i = a_i \times b_i$

a_i	b_i	s_i
0	0	0
0	1	0
1	0	0
1	1	1

При складанні і відніманні в однойменних вагових розрядах в загальному випадку приймають участь три змінних: цифри вагових розрядів a_i, b_i чисел і перенос p_i з попереднього розряду (при складанні) чи позика з попереднього розряду z_i (при відніманні). Результатом складання є розряд суми s_i та перенос p_{i+1} у наступний верхній розряд. Результатом віднімання є різниця $s_i = a_i – b_i - z_i$ та позика z_{i+1} в наступний верхній розряд.

В ЕОМ існує поняття розрядної сітки. Розрядна сітка характеризується кількістю розрядів для представлення числа. Це означає, що всі числа в ЕОМ повинні мати однакову кількість розрядів, в тому числі і результат. Це іноді приводить до того, що результат складання, віднімання виходить за розрядну сітку і у межах розрядної сітки стає неправильним. Таку ситуацію називають переповненням. За звичаєм ця ситуація фіксується у спеціальному додатковому розряді (ознаці) переповнення. Іноді цю ознаку називають переносом з верхнього розряду розрядної сітки.

Приклад 10. Скласти два числа, представлені п’ятьма двійковими розрядами цілої частини та двома розрядами дробової частини.

	Переповнення	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	Дес. сист.
Перенос	1	1	0	1	1	1	1	0	0	
Операнд1		1	1	0	1	1	1	, 1	0	55,50
Операнд2		0	1	0	0	1	1	, 1	1	19,75
Результат	1	0	0	1	0	1	1	, 0	1	11,25

У даному прикладі результат вийшов за межі розрядної сітки, що було зафіксовано у ознаці “переповнення”.

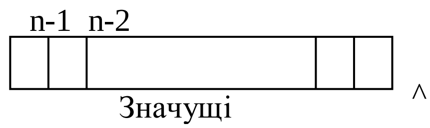
Приклад 11. Відняти два числа, представлені п’ятьма двійковими розрядами цілої частини та двома розрядами дробової частини.

	Переповнення	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	Дес. сист.
Позика	0	0	1	0	1	0	0	1	0	
Операнд1		1	1	1	0	0	1	, 1	0	59,50
Операнд2		0	0	1	0	1	1	, 1	1	13,25
Результат	0	1	0	1	1	1	0	, 0	1	36,25

У даному прикладі переповнення не зафіксовано і результат є правильним.

2.3 Формат з фіксованою комою.

Цілі числа без знаку. Формат цілих двійкових чисел без знака має вид, показаний на мал. 1. Тут значок (^) позначає місце розташування двійкової коми (крапки). Як видно з мал. 1, усі розряди числа є значущими, а двійкова кома знаходиться праворуч, як говорять, фіксована після нижнього значущого розряду. Звідси з'являється ще одна назва цього формату й інших аналогічних форматів – формат з фіксованою комою. Точно так можна двійкову кому розташувати ліворуч від старшого розряду, тоді всі числа будуть дробовими, тобто менше 1.



Мал. 1. Формат цілих

Цілі числа без знаку при програмуванні використовуються для представлення числових об'єктів, що принципово не можуть бути негативними, наприклад, адреси комірок пам'яті, лічильники повторень циклів і т.п.

Цілі числа зі знаком. Щоб комп'ютери могли оперувати позитивними і негативними числами, один з розрядів розрядної сітки комп'ютера застосовується під знак числа S. За звичаєм ним є верхній (лівий) біт, а стандартне кодування знака має вид:

S = 0 - число позитивне

S = 1 - число негативне

Мал. 2. Загальний формат цілих чисел зі знаком.

Таке представлення ще називають прямим кодом. У прямого коду є істотні недоліки з погляду реалізації операції алгебраїчного складання, яка вимагає аналіз знаків операндів і вибір фактичної операції складання чи віднімання.

Щоб усунути цей недолік і спростити електронну реалізацію складання і віднімання алгебраїчних чисел, використовується додавальний код, що дозволяє операцію віднімання замінити операцією складання. При цьому результат виходить у додатковому коді і знак результату формується автоматично.

Додатковий код негативного цілого числа визначається по формулі:

$$A = 2^{n} - |A|, \qquad (4)$$

де n - кількість розрядів числа. Додатковий код позитивного числа збігається з прямим кодом. Для одержання додаткового коду не обов'язково виконувати віднімання. Досить узяти модуль числа, потім усі біти інвертувати, тобто замінити на протилежні, і до отриманого числа додати 1, а в знаковий біт записати 1.

Додатковий код негативного дробового числа визначається по формулі:

$$D = 1 - |A|, \qquad (5)$$

Для віднімання негативних чисел і складання алгебраїчних чисел також використовується зворотний код, який для негативних чисел утворюється шляхом інверсії розрядів значущої частини і запису в розряд знаку 1.

Результат представляється у зворотному коді. При виникненні переповнення для утворення правильного результату одиниця переповнення додається до нижнього розряду числа.

Переповнення для зворотного і для додаткового кодів виникає тільки у випадку складання чисел одного знаку. В разі, якщо знак результату протилежний знаку операндів, то це є ознакою переповнення.

Приклад 12. Виконати складання цілих чисел в додатковому коді 12567+(-4567) числа представити 16 бітами.

```
12567 = 0011000100010111B
-4567 = 1001000111010111B  (прямий код)
        1110111000101000B  (інверсія, знак не інвертується)
      +                1B
        1110111000101001B  (додатковий код)
               0011000100010111B
______________+1110111000101001B
               0001111101000000B =+8000
```

Приклад 13. Виконати складання цілих чисел в зворотному коді 12567+(-4567) числа представити 16 бітами.

```
12567 = 0011000100010111B
-4567 = 1001000111010111B  (прямий код)
        1110111000101000B  (інверсія, знак не інвертується)

               0011000100010111B
______________+1110111000101000B
             1 0001111101110111B
______________+               1B
               0001111101000000B =+8000
```

2.4 Завдання до частини 1.

Відповідно до варіантів з таблиці 5 виконати:

1. Перевести числа з десяткової системи в двійкову систему обчислення з точністю 16 двійкових розрядів у цілій частині і 8 – у дробовій.

2. Перевести числа з двійкової системи обчислення в десяткову.

3. Скласти і відняти цілі числа в двійковій системі обчислення, числа представити 16 розрядами (міняти місцями числа не припустимо).

4. Виконати віднімання з використанням додаткового і зворотнього коду. Числа представити 16 розрядами зі знаком.

5. Перевести числа в базовий одинарний формат, записати в шістнадцятиковій системі і виконати складання.

Примітка: завдання оформлюються за зразком наведених прикладів.

Таблиця 5. Варіанти завдань

N	Завдання 1	Завдання 2	Завдання 3	Завдання 4	Завдання 5
1	10456,17 32007,04	11100010,01001101 10101011,11110001	6756 ± 9876 456 ± 746	6756-9876 456-746	10456,17 32007,04
2	20436,08 31007,12	10101010,11000101 11101011,10110101	3956 ± 4877 856 ± 976	3956-4877 856-976	20436,087 31007,123
3	9436,187 27207,029	11001110,00110101 10001011,10100011	3864 ± 2287 347 ± 593	9927-1387 1245-8714	9436,187 27207,029
4	27476,555 38214,081	10111001,10010111 11000011,00011101	5742 ± 4144 444 ± 502	2328-9021 635-717	27476,555 38214,081
5	19320,701 38048,028	10111110,11001001 10100111,11110001	2876 ± 6452 561 ± 734	6677-9999 4792-3869	32986,311 19463,088
6	41331,492 11037,186	11100110,11001111 10011111,10011111	8934 ± 5322 643 ± 724	4792-3869 162-100	19463,088 22333,675
7	45456,452 45007,063	11000010,01001101 10100111,10100001	3456 ± 5676 534 ± 745	8042-5333 537-999	19320,701 38048,028
8	40436,487 37807,453	10011010,11000101 11100101,10110101	3989 ± 8777 886 ± 767	8900-9988 678-345	27409,622 8564,123
9	6636,587 27207,029	10000011,01100011 10001011,10100011	547 ± 343 347 ± 443	6533-8734 927-998	41331,492 11037,186
10	79656,755 35414,481	10011001,10010111 11000011,00011111	332 ± 444454 ± 589	5678-2349 115-786	34879,478 5555,444
11	34320,701	10111110,11001001	2876 ± 6452	5649-2999	17006,072

	98048,028	10100111,11110001	561 ± 734	829-777	22007,543
12	41331,492 90037,186	11110110,11101111 10010111,10111111	7934 ± 4322 943 ± 324	8763-9678 315-399	18900,008 11672,770
13	22007,043 3284,111	10101001,11110001 10111100,10111101	956 ± 746 7800 ± 8009	9947-8557 835-983	29006,789 11807,003
14	7436,187 17207,029	11101011,10110101 10001011,10100011	356 ± 976 4864 ± 2287	6565-3449 561-989	46578,305 7822,888
15	77476,555 48214,081	10111001,10010111 11000011,00011101	9742 ± 4144 544 ± 502	9097-6382 748-899	7236,666 37307,019
16	89320,701 18048,028	10111110,11001001 10100111,11110001	6876 ± 6452 561 ± 834	8765-3467 511-666	24899,477 10003,555
17	45331,492 11455,586	11100110,11001111 11011111,10011111	5634 ± 5522 645 ± 524	7122-9099 665-810	20876,308 28204,818
18	79645,755 76414,481	10011001,10010111 11000011,00011111	632 ± 444 454 ± 589	4675-8474 564-987	30000,900 8769,222
19	45320,701 98048,028	10111110,11001001 10100111,11110001	6776 ± 6452 561 ± 534	8806-6099 591-834	16371,432 32047,128
20	55320,761 78548,028	10110110,11111001 11100111,10010001	9776 ± 6054 531 ± 634	9849-853 230-896	21479,218 14222,335
21	15327,591 90042,027	10111100,01001001 11110111,11100001	6076 ± 6952 661 ± 537	5078-2049 195-706	39879,408 5755,494
22	22007,043 3284,111	11000010,01001101 10100111,10100001	356 ± 976 4864 ± 2287	4649-3999 809-775	17806,073 22807,843
23	7436,187 17207,029	10011010,11000101 11100101,10110101	9742 ± 4144 544 ± 502	8793-9678 335-399	18910,018 17672,570
24	77476,555 48214,081	10000011,01100011 10001011,10100011	6876 ± 6452 561 ± 834	9247-8007 805-903	23106,779 21807,103
25	89320,701 18048,028	10011001,10010111 11000011,00011111	5634 ± 5522 645 ± 524	6065-3349 541-989	42568,345 7022,888
26	45331,492 11455,586	10111110,11001001 10100111,11110001	632 ± 444 454 ± 589	9007-6382 798-999	7836,666 30307,019
27	64382,356 22189,057	11001101,1111101 11111110,1111101	872±949 3900±722	4444-2203 6543-8888	72723,721 988,077
28	77145,208 11111,111	10101010,010101 10010010,001111	5656± 4545 565±454	3358-5348 904-567	12386,654 5548,648
29	21675,548 19468,358	11100011,101011 11000011,100001	29988±654 5848±67	6764-8435 5645-5640	5854,885 215,948
30	21548,448 58964,333	11111001,111001 10000111,001111	3879± 2222 679±976	9431-6666 379±387	4986,654 489,484

3 Частина 2. Алгоритми множення та ділення двійкових чисел

Мета роботи: придбання практичних навичок множення та ділення двійкових чисел в прямому, зворотному і додатковому кодах.

3.1 Загальні положення

Студентам слід знати, що найбільш відомими і простими є два основних способи виконання операції множення чисел у формі з фіксованою комою:

- множення з молодших розрядів множника;
- множення зі старших розрядів множника.

В обох випадках операція множення складається з послідовних операцій додавання часткових добутків. Виконання операції додавання залежить від значень розрядів множника: якщо деякий розряд множника дорівнює одиниці, то до суми часткових творів додається множене; якщо розряд множника дорівнює нулю, то множене не додається.

Нехай А = 0, а 1, а 2, ..., а n - множне в прямому коді (А ≥ 0); В = 0, а 1, а 2, ..., а n - множник в прямому коді (В ≥ 0); З - твір. Тоді С = 0, с1, с2, ..., с2n і

$$C = A \cdot B = A \cdot в_1 \cdot 2^{-1} + A \cdot в_2 \cdot 2^{-2} + ... + A \cdot в_n \cdot 2^{-n}$$

Доданок С, з огляду на можливості зсуву множимо або суми часткових творів, то, можливо обчислено по одному з наступних чотирьох алгоритмів.

Алгоритм 1. Множення з молодших розрядів множника із зсувом суми часткових творів вправо:

$$C = A \cdot B = (...((0 + A \cdot в_n) \cdot 2^{-1} + A \cdot в_{n-1}) \cdot 2^{-1} + ... + A \cdot в_1) \cdot 2^{-1}$$

Множене А послідовно множиться на розряди множника Bn, Bn-1, ..., в1 і додається на кожному кроці до зсунутого вправо на 1 розряд (помноженої на 2-1) сумі часткових творів.

Алгоритм 2. Множення з молодших розрядів множника із зсувом множимо вліво при нерухомій сумі часткових творів:

$$C = A \cdot B = 0 + (2^{-n} \cdot A) \cdot в_n + (2^1 (2^{-n} \cdot A)) \cdot в_{n-1} + (2^1 (2^1 (2^{-n} \cdot A))) \cdot в_{n-2} + ... + (2^{-1} \cdot A) \cdot в_1 .$$

Множене А, спочатку зрушене на n розрядів вправо (помножене на 2-n), множиться послідовно на розряди множника Bn, Bn-1, ..., в1 і після складання з сумою часткових творів на кожному кроці зсувається вліво на 1 розряд (множиться на 21).

Алгоритм 3. Множення зі старших розрядів множника із зсувом суми часткових творів вліво при нерухомому множимо:

$$C = A \cdot B = (...((0 + 2^{-n} \cdot A \cdot в_1) \cdot 2^1 + 2^{-n} \cdot A \cdot в_2)2^1 + ... + 2^{-n} \cdot A \cdot в_{n-1})2^1 + 2^{-n} \cdot A \cdot в_n \ .$$

Множене А, зрушене вправо на n розрядів, множиться послідовно на розряди множника в1, в2, ..., Вn і додається на кожному кроці до молодших розрядів суми часткових творів, яка після цього зсувається вліво на 1 розряд (крім останнього кроку, коли проводиться множення на Вn).

Алгоритм 4. Множення зі старших розрядів множника із зсувом множимо вправо при нерухомій сумі часткових творів:

$$C = A \cdot B = 0 + (2^{-1} \cdot A) \cdot в_1 + (2^{-1}(2^{-1} \cdot A)) \cdot в_2 + ... + (2^{-n} \cdot A) \cdot в_n \ .$$

Множене А на кожному кроці зсувається вправо на 1 розряд, множиться на відповідний розряд множника в1, в2, ..., Вn і додається до суми часткових творів.

Множення чисел, представлених в додатковому і зворотному кодах, має такі важливі особливості:

а) в разі позитивного множника $B = B_{дк} = B_{ок} = 0,в_1в_2...в_n$ при множенні отримуємо відразу додатковий (зворотний) код твори:

$$C_{дк} = A_{дк} \cdot 0,в_1в_2...в_n = (A \cdot B)_{дк},$$

$$\boldsymbol{C_{ок} = A_{ок} \cdot 0,в_1в_2...в_n = (A \cdot B)_{ок}};$$

б) в разі негативного множника $B = B_{дк} = B_{ок} = 1,\tilde{в}_1\tilde{в}_2...\tilde{в}_n$ додатковий (зворотний код твори виходить після введення однієї (двох) корекції:

$$C_{дк} = A_{дк} \cdot 0,\tilde{в}_1\tilde{в}_2...\tilde{в}_n - A_{дк},$$

$$C_{ок} = A_{ок} \cdot 0,\tilde{в}_1\tilde{в}_2...\tilde{в}_n - A_{ок} + A_{ок} \cdot 2^{-n};$$

в) множення коду множимо на цифрові розряди коду множника виконується за допомогою одного з розглянутих алгоритмів, при цьому операції додавання і зсуву виконуються у відповідному коді.

Множення чисел, представлених у формі з плаваючою комою, включає наступні етапи:

а) множення мантис, як чисел з фіксованою комою, по одному з алгорітмов1-4;

б) складання порядків;

в) нормалізація мантиси з відповідною корекцією порядку результату.

Студентам слід звернути увагу на те, що при множенні чисел з плаваючою комою можливе переповнення розрядної сітки.

Приклад 14. x = 0.6 та y = 0.75 перемножити за алгоритмом 3 у прямому коді

$x= 0.1001 = x_{ok}=x_{dk}$

$y= 0.1100 = y_{ok}=y_{dk}$

$-x = 1.1001$

$-y = 1.1100$

$-x_{ok}=11.01101111$

$-x_{dk}=11.01110000$

$-y_{ok}=11.10011111$

$-y_{dk}=11.01000000$

1. З/ПК x*y

x=Мн=0,10010000

y=Мт=0,11000000

РгМт	РгМн	Сума	Счт
1100	00001001	00000000 00001001	0
1000	00001001	00001001 00010010 00011011	1
0000	00001001	00011011 00110110 00000000	2
0000	00001001	00110110 01101100 00000000	3
		01101100	

При діленні чисел з фіксованою комою в ЕОМ найбільш поширеним є метод, заснований на вирахуванні подільника з діленого. При цьому для поділу застосовуються два алгоритму:

а) алгоритм ділення без відновленням залишку;

Крок 1. Визначення основної операції (таблиця 6);

Крок 2. Виконання пробного віднімання: над дільним виконати основну операцію, перевірити отриманий залишок, якщо залишок має такий саме знак, що і дільне, то ділення неможливо, завершення операції, інакше, якщо знаки різні то переходимо до кроку 3.

Крок 3. Виконання операції протилежної основній (якщо основна операція 2R + (-ДТ), то протилежність буде 2R + ДТ, операція 2R – це зсув залишка вліво на один розряд), перехід на крок 4.

Крок 4. Визначення цифри частки, по даним таблиці 6. Перехід на крок 5.

Крок 5. Визначення наступної дії: якщо знак отриманого залишка та дільного співпадають то перехід на крок 3., якщо не співпадають перехід на крок 6.

Крок 6. Виконання основної операції, перехід на крок 4.

б) алгоритм розподілу з відновлення залишку.

крок 1. Визначення основної операції (таблиця 6);

крок 2. Виконання пробного віднімання: над дільним виконати основну операцію, перевірити отриманий залишок, якщо залишок має такий саме знак, що і дільне, то ділення неможливо, завершення операції, інакше, якщо знаки різні то переходимо до кроку 3.

Крок 3. Відновлення залишку: виконання операції протилежної основній без зсуву залишка вліво на один розряд, перехід на крок 4.

Крок 4. Виконання основної операції. Перехід до кроку 5.

Крок 5. Визначення цифри частки, по даним таблиці 6. Перехід на крок 5.

Крок 6. Визначення наступної дії: якщо знак отриманого залишка та дільного співпадають то перехід на крок 3., якщо не співпадають перехід на крок 4.

Задля визначення необхідної операції необхідно визначати основну операцію та значення частки, дані наведені в таблиці 6.

Таблиця 6 – Правила ділення двійкових чисел

Знак Дм (дільне)	+	+	-	-
Знак Дт (дільник)	+	-	+	-
Основна операція	2R + (-ДТ)	2R + ДТ	2R + ДТ	2R + (-ДТ)
Визначення цифри частки	R < 0, ЧТ = 0 R > 0, ЧТ = 1	R < 0, ЧТ = 0 R > 0, ЧТ = 1	R < 0, ЧТ = 1 R > 0, ЧТ = 0	R < 0, ЧТ = 1 R > 0, ЧТ = 0

Де R – поточний залишок

3.2 Завдання до частини 2

1. Використовуючи дані таблиці 7, обчислити добуток чисел ± x і ± y з подвійною точністю, якщо довжина розрядної сітки операндів без урахування знакових розрядів дорівнює 4-м двійковим розрядам. Процес обчислень потрібно представити в таблицях наступного виду:

Логічна умова				
	Мт	Мн	СМ	СчТ

Після обчислень перевірити правильність отриманих результатів.

2. Використовуючи дані таблиці 8 і 9, розділити (± X) і (± Y). Процес обчислень представити в трасувань таблицях виду:

Логічна умова	Стан операційного елемента				
	СМ	Чт	СчТ	Дм	Дт

Таблиця 7 - Варіанти завдань до множення

№ п/п	*x*	*y*	+/+	+/-	-/+	-/-
1	0.15	0.96	1/ПК	2/МЗК	3/МДК	4/МДК
2	0.17	0.84	2/ПК	3/МДК	4/МЗК	1/МЗК
3	0.35	0.88	3/ПК	4/МЗК	1/МДК	2/МДК
4	0.33	0.79	4/ПК	1/МДК	2/МЗК	3/МЗК
5	0.47	0.69	1/ПК	2/МЗК	3/МДК	4/МДК
6	0.77	0.63	2/ПК	3/МДК	4/МЗК	1/МЗК
7	0.81	0.45	3/ПК	4/МЗК	1/МДК	2/МДК
8	0.56	0.99	4/ПК	1/МДК	2/МЗК	3/МЗК
9	0.68	0.37	1/ПК	2/МЗК	3/МДК	4/МДК
10	0.19	0.83	2/ПК	3/МДК	4/МЗК	1/МЗК
11	0.99	0.27	3/ПК	4/МЗК	1/МДК	2/МДК
12	0.11	0.95	4/ПК	1/МДК	2/МЗК	3/МЗК
13	0.90	0.35	1/ПК	2/МЗК	3/МДК	4/МДК
14	0.30	0.79	2/ПК	3/МДК	4/МЗК	1/МЗК

15	0.97	0.07	3/ПК	4/МЗК	1/МДК	2/МДК
16	0.29	0.81	4/ПК	1/МДК	2/МЗК	3/МЗК
17	0.65	0.75	1/ПК	2/МЗК	3/МДК	4/МДК
18	0.26	0.81	2/ПК	3/МДК	4/МЗК	1/МЗК
19	0.57	0.69	3/ПК	4/МЗК	1/МДК	2/МДК
20	0.77	0.79	4/ПК	1/МДК	2/МЗК	3/МЗК
21	0.72	0.45	1/ПК	2/МЗК	3/МДК	4/МДК
22	0.94	0.93	2/ПК	3/МДК	4/МЗК	1/МЗК
23	0.59	0.81	3/ПК	4/МЗК	1/МДК	2/МДК
24	0.64	0.97	4/ПК	1/МДК	2/МЗК	3/МЗК
25	0.82	0.43	1/ПК	2/МЗК	3/МДК	4/МДК
26	0.44	0.77	1/ПК	2/МЗК	3/МДК	4/МДК
27	0.22	0.82	2/ПК	3/МДК	4/МЗК	1/МЗК
28	0.34	0.56	3/ПК	4/МЗК	1/МДК	2/МДК
29	0.86	0.27	4/ПК	1/МДК	2/МЗК	3/МЗК
30	0.79	0.98	1/ПК	2/МЗК	3/МДК	4/МДК

Таблиця 8 – Визначення алгоритму ділення

SgX	*SgY*	КОД	Алгоритм
+	+	МДК	БВЗ
+	-	МОК	ЗВЗ
-	+	МОК	БВЗ
-	-	МДК	ЗВЗ

Таблиця 9 - Варіанти завдань до ділення

№ п/п	X	Y	№ п/п	X	Y
1	0,1000	0,1001	16	0,1000	0,1011
2	0,1001	0,1010	17	0,1001	0,1100
3	0,1010	0,1011	18	0,1010	0,1101
4	0,1011	0,1100	19	0,1011	0,1110
5	0,1100	0,1101	20	0,1100	0,1111
6	0,1101	0,1110	21	0,1000	0,1100
7	0,1110	0,1111	22	0,1001	0,1101
8	0,1000	0,1010	23	0,1010	0,1110
9	0,1001	0,1011	24	0,1011	0,1111

10	0,1010	0,1100	25	0,1000	0,1101
11	0,1011	0,1101	26	0,1001	0,1110
12	0,1100	0,1110	27	0,1010	0,1111
13	0,1101	0,1111	28	0,1001	0,1101
14	0,1100	0,1110	29	0,0111	0,1001
15	0,0110	0,1100	30	0,1011	0,1110

Приклад 15. x/y

Дм: $x_{мдк}$= 00.1100

Дт: $y_{мдк}$= 00.1110

-Дт: -$y_{мдк}$=11.0010

Основна операція: $2R_{i-1}$ + (-Дт)

Логічна умова	Стан операційного елемента				
	СМ	Чт	Сч	Дм	Дт
Пробне віднімання	00.1100 11.0010	00.0000	5	00.1100	00.1110
Зсув залишку вліво Додавання протилежного значення	11.1110 11.1100 00.1110				
Визначення цифри частки Зсув залишку Виконання основної операції	00.1010 01.0100 11.0010	00.000$\underline{1}$	4		
Визначення цифри частки Зсув залишку Виконання основної операції	00.0110 00.1100 11.0010	00.001$\underline{1}$	3		
Визначення цифри частки Зсув залишку Виконання операції протилежної основній	11.1110 11.1100 00.1110	00.011$\underline{0}$	2		
Визначення цифри частки Зсув залишку Виконання основної операції	00.1010 01.0100 11.0010	00.110$\underline{1}$	1		
	00.0110	00.1101$\underline{1}$	0		

3.3 Вимоги до змісту звіту

Звіт розрахунково-графічної роботи повинен мати наступні розділи:

1. Постановка завдання (загальна і для конкретного варіанту).
2. Результати виконання прикладів, що наведено в таблицях, як в прикладах.
3. Висновки.

Список рекомендованої літератури

1. Ашарина, И.В. Основы программирования на языках С и С++ : курс лекц. / И.В. Ашарина. – М. : Горячая Линия-Телеком, 2012. – 208 с.
2. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. – М. : Горячая Линия-Телеком, 2012. – 208 с.
3. Дорогов, В.Г. Основы программирования на языке С : учеб. пособ. для вуз. / В.Г. Дорогов, Е.Г. Дорогова ; [под общ. ред. проф. Л.Г. Гагарина]. – М. : Форум ; ИНФРА-М, 2013. – 224 с.
4. Колдаев, В.Д. Основы алгоритмизации и программирования : учеб. пособ. / В.Д. Колдаев ; [под общ. ред. проф. Л.Г. Гагарина]. – М. : Форум ; ИНФРА-М, 2012. – 416 с.
5. Культин, Н.Б. Основы программирования в Turbo C++ / Н.Б. Культин. – СПб. : BHV, 2013. – 464 с.
6. Семакин, И.Г. Основы алгоритмизации и программирования : учеб. для студ. учреждений сред. проф. образов. / И.Г. Семакин, А.П. Шестаков. – М. : ИЦ Академия, 2012. – 400 с.
7. Черпаков, И.В. Основы программирования : учеб. и практикум для прикладного бакалавриата / И.В. Черпаков. – Люберцы : Юрайт, 2016. – 219 с.
8. Черпаков, И.В. Основы программирования : учеб. и практикум для СПО / И.В. Черпаков. – Люберцы : Юрайт, 2016. – 219 с.