

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Організація циклічних процесів. Складні цикли»

Варіант 30

Виконав студент ІІ-12 Тарасюк Євгеній Сергійович

Перевірив _____

Київ 2021

Лабораторна робота 5.

Організація циклічних процесів. Складні цикли.

Мета: дослідити особливості роботи складних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

Задача 30 (варіант 30).

30. Дано натуральне число m . Знайти таке натуральне число n , що двійковий запис n отримується із двійкового запису m зміною порядку цифр на зворотний.

Розв'язок.

1. Постановка задачі.

Початкові дані - це натуральне число, додаткових змінних для розв'язку не потрібно. Додаткових умов не потрібно. Для обчислення використовуватимемо два ітеративних цикли: перший - знаходження результату в двійковій формі способом ділення, другий - переведення його в десяткову форму. Результатом розв'язку є натуральне число. Використовуватимемо стандартні логічні та арифметичні операції (ступінь числа позначаємо як $^$, цілочисельне ділення як $/$, знаходження остачі як $\%$).

2. Побудова математичної моделі

Таблиця змінних та функцій:

Змінні	Тип	Ім'я	Призначення
Число num	Натуральне число	num	Збереження початкових даних

Результат	Натуральне число	res	Збереження результату
Двійкова форма	Натуральне число	binary	Збереження результату в двійковій формі
Степінь двох	Натуральне число	power	Допоміжна змінна для переведення результату в десяткову форму

3. Псевдокод алгоритму

Крок 1

Початок

Введення *num*

{Пошук результату в двійковій формі}

{Переведення результату в десяткову форму}

Виведення *res*

Кінець.

Крок 2

Початок

Введення *x*

binary = 0

Повторити поки *num* > 0

binary = *binary* * 10

binary = *binary* + *num* % 2

num = *num* // 2

Все повторити

{Переведення результату в десяткову форму}

Кінець

Крок 3

Початок

Введення *x*

binary = 0

Повторити поки *num* > 0

binary = *binary* * 10

binary = *binary* + *num* % 2

num = *num* // 2

Все повторити

power = 0

res = 0

Повторити поки *binary* > 0

Якщо *binary* % 2 != 0

res = *res* + 2 ^ *power*

Все якщо

power = *power* + 1

binary = *binary* // 10

Все повторити

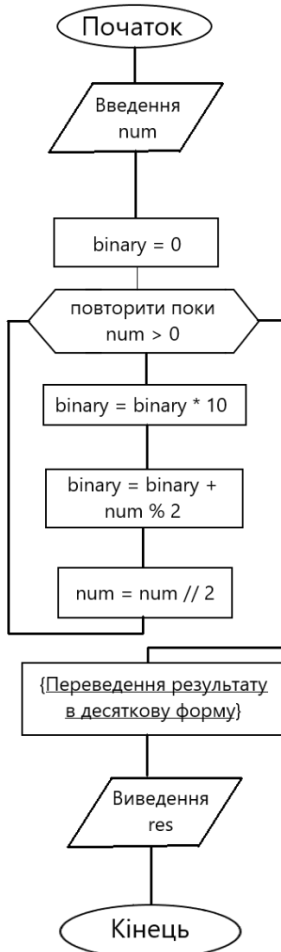
Кінець

4. Блок схема алгоритму

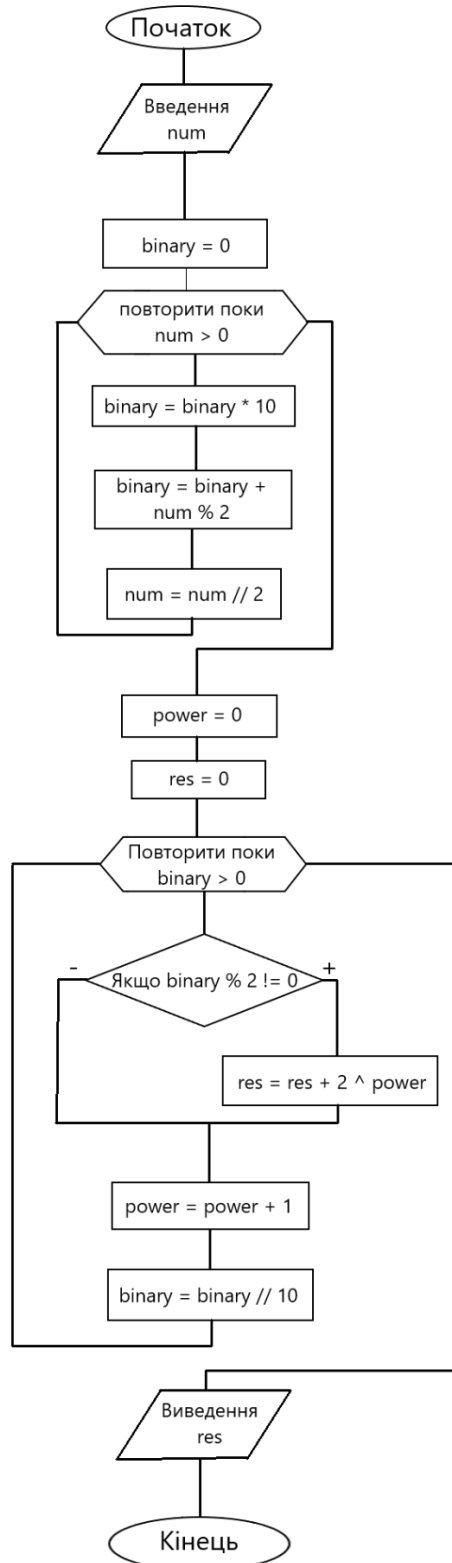
Крок 1



Крок 2



Крок 3



5. Випробування алгоритму.

Перевіримо правильність алгоритму для різних вхідних даних:

	<i>Тест 1</i>	<i>Тест 2</i>	<i>Тест 3</i>
Введення num	num = 32	num = 63	num = 54
Повторити поки num > 0 (початок ітерації 1)	num = 32 binary = 0	num = 63 binary = 0	num = 54 binary = 0
Початок ітерації 2:	num = 16 binary = 0	num = 63 binary = 0	num = 27 binary = 0
Початок ітерації 3:	num = 8 binary = 0	num = 31 binary = 1	num = 13 binary = 1
Початок ітерації 4:	num = 4 binary = 0	num = 15 binary = 11	num = 6 binary = 11
Початок ітерації 5:	num = 2 binary = 0	num = 7 binary = 111	num = 3 binary = 110
Початок ітерації 6:	num = 1 binary = 0	num = 3 binary = 1111	num = 1 binary = 1101
Початок ітерації 7:	-	num = 1 binary = 11111	-
Все повторити	num = 0 binary = 1	num = 0 binary = 111111	num = 0 binary = 11011
Повторити поки binary > 0 (початок ітерації 1)	binary = 1 res = 0	binary = 111111 res = 0	binary = 11011 res = 0
Початок ітерації 2:	-	binary = 11111 res = 1	binary = 1101 res = 1
Початок ітерації 3:	-	binary = 1111 res = 3	binary = 110 res = 3

Початок ітерації 4:	-	binary = 111 res = 7	binary = 11 res = 3
Початок ітерації 5:	-	binary = 11 res = 15	binary = 1 res = 1
Початок ітерації 6:	-	binary = 1 res = 31	-
Все повторити	binary = 0 res = 1	binary = 0 res = 63	binary = 0 res = 27
Виведення res	1	63	27

Висновки

Було досліджено особливості роботи складних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.