An Interactive C Project to Explore and Understand Number Systems

LEARN NUMBER SYSTEMS WITH NUMCRAFT

LEARN, PRACTICE THE NUMBER SYSTEMS IN C

+91 9370409503

rehanmokashi786@gmail.com

rehan-mokashi-7b32472a2

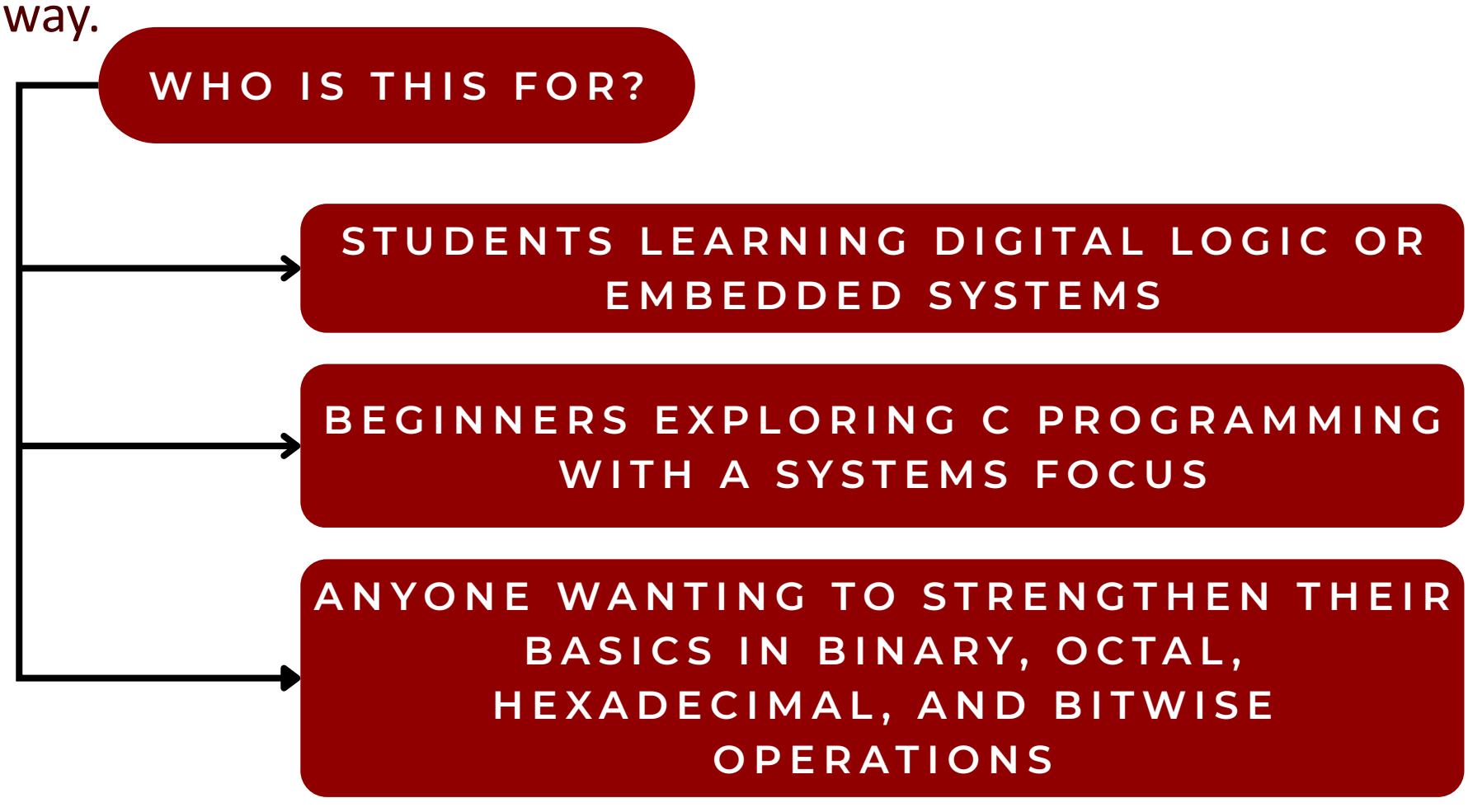
rmokashi01

REHAN MOKASHI (B. TECH - CSE)
JULY 2025
TRAINEE AT EMERTXE INFORMATION TECHNOLOGIES, BENGALURU

1. Introduction

NumCraft: Learn Number Systems is an interactive command-line-based C project designed to simplify and visualize complex number system concepts. From beginners to intermediate learners, this tool offers a hands-on learning experience by integrating quizzes, simulations, and visual aids all built in pure C.

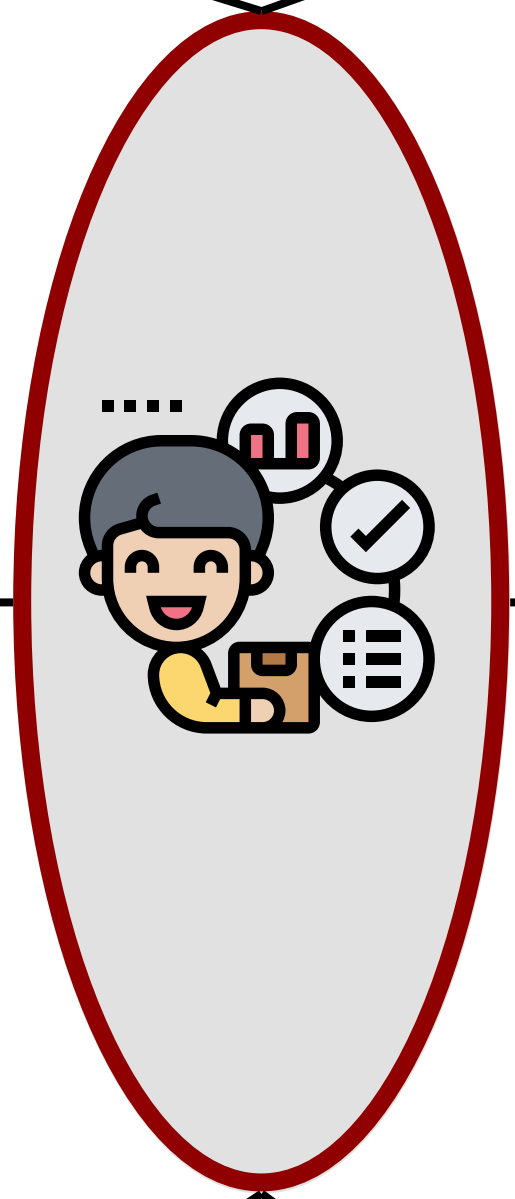
Understanding number systems is a foundational skill in computer science, digital electronics, and low-level programming. However, for many learners, grasping concepts like binary conversions, two's complement, or bitwise operations can be intimidating. NumCraft aims to break this barrier by offering an engaging and intuitive platform to explore these topics practically. Through different learning modules — such as quizzes, conversion visualizers, two’s complement simulations, and bit-level manipulators — learners can experiment, analyze, and reinforce their understanding in a structured yet enjoyable way.



2. C Concepts Applied in This Project

POINTERS AND MEMORY

STRUCTURES



CONDITIONAL AND LOOPING CONSTRUCTS

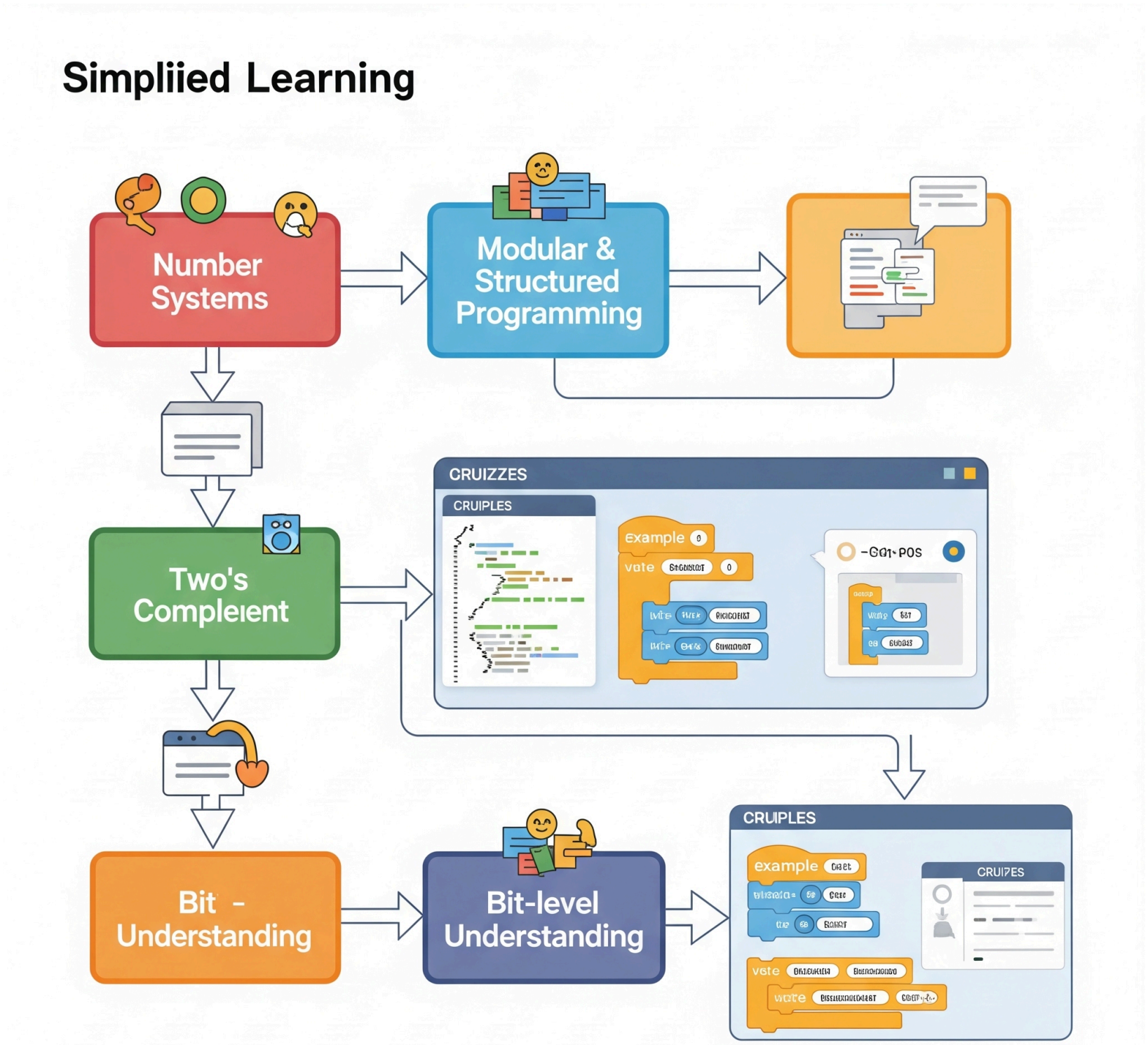
BIT MANIPULATION

COMMAND-LINE ARGUMENTS

CHARACTER AND STRING HANDLING

3. Project Objectives

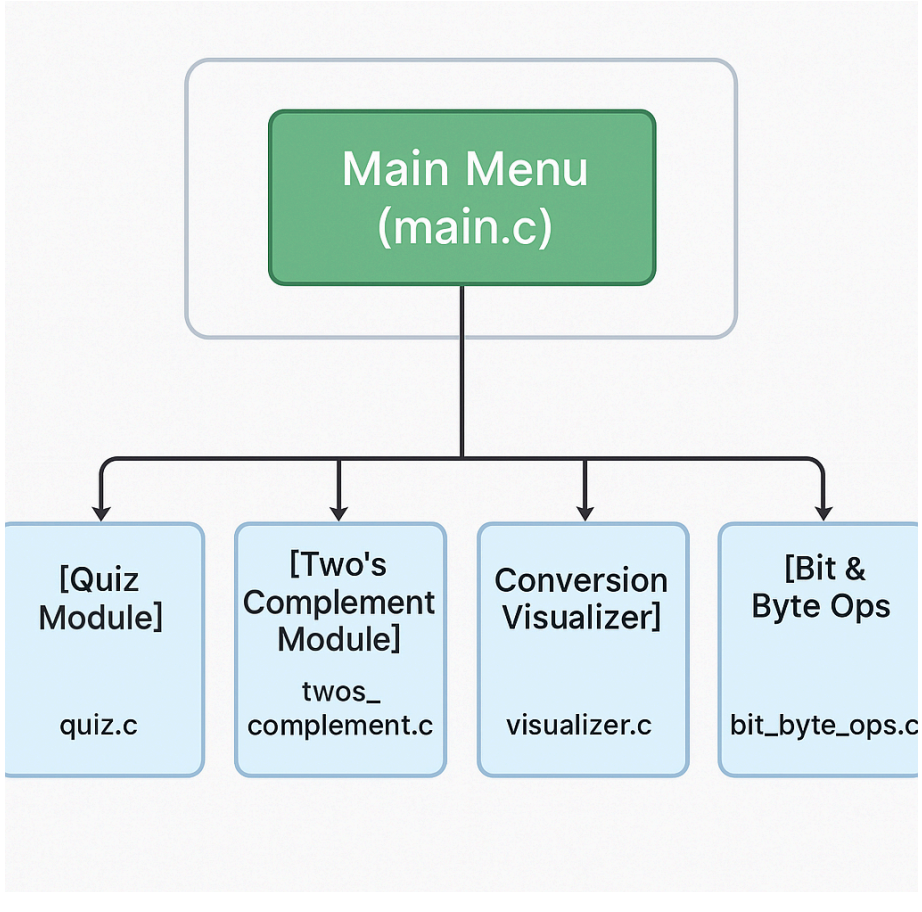
1. **Simplify Learning of Number Systems**
2. **Practice Modular and Structured Programming in C**
3. **Develop Intuition Behind Two's Complement**
4. **Enable Bit-Level Understanding**
5. **Make Learning Fun and Interactive**
6. **Bridge Theory and Practical Coding Skills**

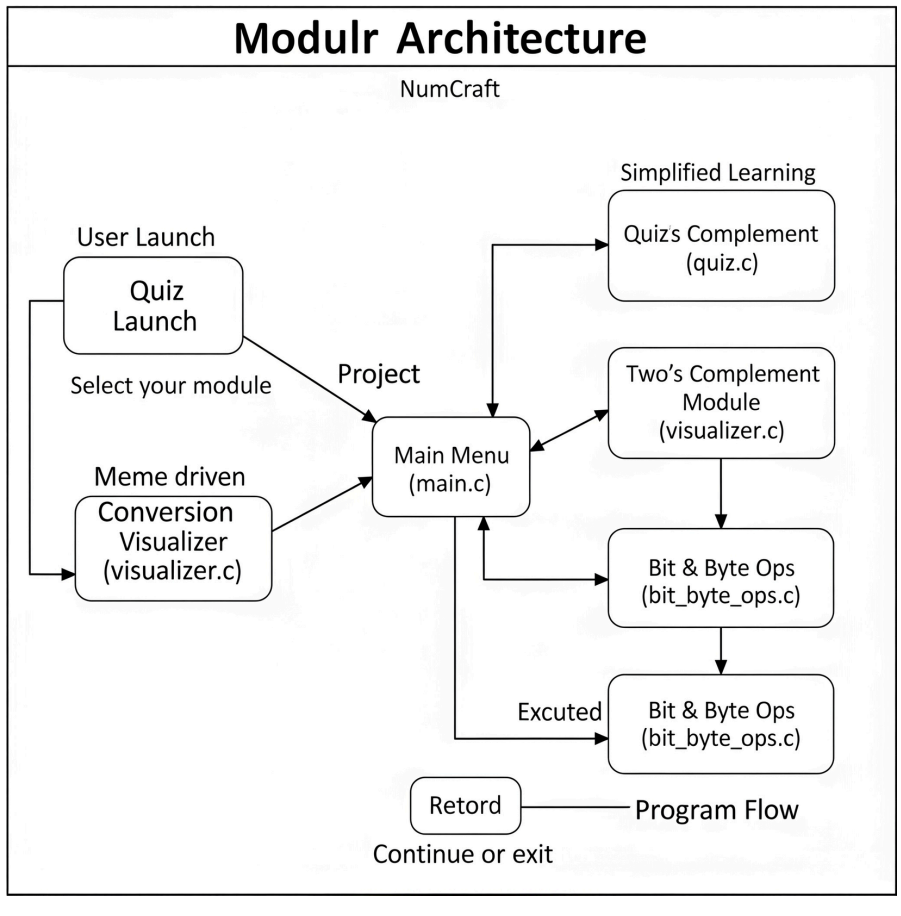


4. File Organization

FILE NAME DESCRIPTION

1. **main.c :**
    i. **Entry point of the program. Displays main menu and handles user input.**
2. **quiz.c**
    i. **Handles the number system quiz logic.**
3. **twos_complement.c**
    i. **Simulates two's complement operation for signed integers.**
4. **visualizer.c**
    i. **Converts between binary, decimal, octal, and hexadecimal values.**
5. **bit_byte_ops.c**
    i. **Performs bitwise operations and shows visual effects on bits.**
6. **input_utils.c**
    i. **Helper functions to take safe user input.**
7. **include/*.h**
    i. **Header files containing function declarations and constants.**
8. **build/ Output directory for compiled binaries.**





5. User Interface Design

5.1 Main Menu Design

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
========================================
     Number System Learning Project
========================================

Main Menu:
1. Number Conversion Quiz Game
2. Two's Complement Arithmetic Simulator
3. Number Conversion Visualizer
4. Bit/Byte Conversion & Bitwise Operations
5. Exit

Enter your choice: _
```

5.2 Quiz Module Interface (Random Generation)

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
========================================
     Number System Learning Project
========================================

Main Menu:
1. Number Conversion Quiz Game
2. Two's Complement Arithmetic Simulator
3. Number Conversion Visualizer
4. Bit/Byte Conversion & Bitwise Operations
5. Exit

Enter your choice: 1

==========================================
   🎯  Number Conversion Quiz Game 🎯
==========================================
 Test your skills on binary, decimal, hex
 conversions and arithmetic operations!
------------------------------------------
Q: What is 1011 + 1 (binary)? 1100
✅ Correct!
Score: 1/1
Do you want to continue? (y/n): y
Q: What is 111100 (binary) in decimal? 50
❌ Incorrect.
Score: 1/2
Do you want to continue? (y/n):
```

5.3 Two’s Complement Simulator Interface.

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
     Number System Learning Project
=======================================

Main Menu:
1. Number Conversion Quiz Game
2. Two's Complement Arithmetic Simulator
3. Number Conversion Visualizer
4. Bit/Byte Conversion & Bitwise Operations
5. Exit

Enter your choice: 2

==========================================
   🧮 Two's Complement Arithmetic Simulator
==========================================
 Understand how signed binary arithmetic
 works using two's complement logic.
------------------------------------------

Choose operation:
1. Addition (a + b)
2. Subtraction (a - b)
Enter choice: 1
Enter bit size (e.g. 4, 8, 16): 8
Enter first number (a): 10
Enter second number (b): 15

Bit-by-bit Addition:
  A      : 00001010
  B      : 00001111
  --------
  Result : 00011001
  Carry  : 00011100

Summary:
----------------------------------
  Decimal:    10 + 15 = 25
  Binary A:   00001010
  Binary B:   00001111
  Result:     00011001 (Decimal: 25)
  Carry Bits: 00011100
----------------------------------

Do you want to try another operation? (y/n):
```

- **Visually shows each step of the conversion**
- **Displays both binary form and final signed value**

5.4.1 Conversion Visualizer Interface

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
========================================
     🔢  Number Conversion Visualizer
========================================
1. Decimal → Binary
2. Binary → Decimal
3. Decimal → Hexadecimal
4. Hexadecimal → Decimal
5. Decimal → Octal
6. Octal → Decimal
7. Back to Main Menu
Enter your choice: 1
Enter a decimal number: 10

========================================
   🔢  Decimal to Binary - Visualizer
========================================
  This shows how decimal numbers convert
  step-by-step into binary using division.
----------------------------------------

🔢 Step-by-Step Division Table:
```

Quotient	Remainder
10	0
5	1
2	0
1	1
0	-

```
🔢  Final Binary for Decimal 10 (MSB ← LSB): 1010 🔢
```

5.4.2 Binary to Decimal

```
7. Back to Main Menu
Enter your choice: 2
Enter a binary number: 1111

================================================
   🔢 Binary to Decimal - Visualizer
================================================
  Shows how each bit contributes to the total.
------------------------------------------------

Binary Input: 1111

Position (2^n):  2^3  2^2  2^1  2^0
Bit Value      :   1    1    1    1
------------------------------------------------
Calculation    :  (1x8) + (1x4) + (1x2) + (1x1) = 15 🔢
```

5.4.3 Decimal to Hexadecimal

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
========================================
     🔢  Number Conversion Visualizer
========================================
1. Decimal → Binary
2. Binary → Decimal
3. Decimal → Hexadecimal
4. Hexadecimal → Decimal
5. Decimal → Octal
6. Octal → Decimal
7. Back to Main Menu
Enter your choice: 3
Enter a decimal number: 14

============================================
   🔢 Decimal to Hexadecimal - Visualizer
============================================
  Shows how repeated /16 gives hex digits.
--------------------------------------------

Decimal Input: 14

Step | Quotient | Remainder | Hex Digit
------------------------------------
  1  |     14    |     14      |     E
  2  |      0    |      -      |     -

🔢  Hexadecimal (MSB ← LSB): E 🔢
```

5.4.4 Hexadecimal to Decimal

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
7. Back to Main Menu
Enter your choice: 4
Enter a hexadecimal number: FF

============================================
   🔢 Hexadecimal to Decimal - Visualizer
============================================
  Shows place value of each hex digit (16ⁿ).
--------------------------------------------

Hex Input: FF

Position (16^n):   16^1  16^0
Hex Digit       :    F     F
Decimal Value   :    240   15
--------------------------------------------
Total Decimal  = (15×16) + (15×1) = 255 🔢
```

5.4.5 Decimal to Octal

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
========================================
     🔢  Number Conversion Visualizer
========================================
1. Decimal → Binary
2. Binary → Decimal
3. Decimal → Hexadecimal
4. Hexadecimal → Decimal
5. Decimal → Octal
6. Octal → Decimal
7. Back to Main Menu
Enter your choice: 5
Enter a decimal number: 17

============================================
   🔢 Decimal to Octal - Visualizer
============================================
  Shows how repeated /8 gives octal digits.
--------------------------------------------

Decimal Input: 17

Step | Quotient | Remainder | Octal Digit
----------------------------------------
  1  |    17    |     1     |    1
  2  |     2    |     2     |    2
  3  |     0    |     -     |    -

🔢  Octal (MSB ← LSB): 21 🔢
```

5.4.6 Octal to Decimal

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
7. Back to Main Menu
Enter your choice: 6
Enter an octal number: 15

================================================
   🔢 Octal to Decimal - Visualizer
================================================
  Shows how each digit contributes (8ⁿ).
------------------------------------------------

Octal Input: 15

Position (8^n):     8^1   8^0
Octal Digit   :      1     5
Decimal Value :        8     5
------------------------------------------------
Total Decimal = (1×8) + (5×1) = 13 🔢
```

5.5 Bit/Byte Conversion & Bitwise Operations

Decimal to bit/byte/kb/mb/gb

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
========================================
     Number System Learning Project
========================================

Main Menu:
1. Number Conversion Quiz Game
2. Two's Complement Arithmetic Simulator
3. Number Conversion Visualizer
4. Bit/Byte Conversion & Bitwise Operations
5. Exit

Enter your choice: 4

============================================
     🔢 Bit/Byte & Bitwise Operations
============================================
1. Bit/Byte/Kilobyte Converter
2. Bitwise Operation Visualizer (AND, OR, etc.)
3. Back to Main Menu
Enter your choice: 1

========================================
       🔢  Bit / Byte Unit Converter
========================================
 Converts between bits, bytes, KB, MB, GB.
----------------------------------------
Enter the value (numeric): 8
Enter the unit (bit/byte/kb/mb/gb): byte

🔢  Conversion Results:
Bits   : 64
Bytes  : 8.000
KB     : 0.007812
MB     : 0.000008
GB     : 0.000000

============================================
     🔢 Bit/Byte & Bitwise Operations
============================================
1. Bit/Byte/Kilobyte Converter
2. Bitwise Operation Visualizer (AND, OR, etc.)
3. Back to Main Menu
Enter your choice:
```

Bitwise Operation Visualizer (AND, OR, etc.)

```
rehan@DESKTOP-5T1VNH6: /mnt/c/Users/REHAN/Desktop/Exertxe/Advance_C/Project 1/number_system_project
==========================================
     🔢 Bit/Byte & Bitwise Operations
==========================================
1. Bit/Byte/Kilobyte Converter
2. Bitwise Operation Visualizer (AND, OR, etc.)
3. Back to Main Menu
Enter your choice: 2

==========================================
      🔢  Bitwise Operation Visualizer
==========================================
 Try AND, OR, XOR, NOT, <<, >> in binary
------------------------------------------
Enter number A: 25
Enter number B: 10

Choose operation:
1. AND (a & b)
2. OR  (a | b)
3. XOR (a ^ b)
4. NOT (~a)
5. Left Shift  (a << 1)
6. Right Shift (a >> 1)
Enter choice: 1

  00011001 (A)
& 00001010 (B)
= 00001000 → 8

Do you want to try another? (y/n): y
Enter number A: 15
Enter number B: 14

Choose operation:
1. AND (a & b)
2. OR  (a | b)
3. XOR (a ^ b)
4. NOT (~a)
5. Left Shift  (a << 1)
6. Right Shift (a >> 1)
Enter choice: 2

  00001111 (A)
| 00001110 (B)
= 00001111 → 15

Do you want to try another? (y/n):
```

thank YOU

Every bit counts, every byte matters — and with each logic built in C, we craft clarity from complexity

- Rehan Mokashi