Programming Assignment 4: Hexadecimal Arithmetic

D1189290 許博琮 Terry

Part 1:

Problem Statement:

The task is to create a C program that adds two hexadecimal numbers. The program takes an even number of hexadecimal digit strings, each with a maximum of 64 bits and no leading zeros, as input. The process continues until both input hexadecimal numbers are "0 0". In each iteration, the program reads two hexadecimal digit strings, performs addition on them, and prints the numbers (`n1`, `n2`), their sum (`sum`), aligned to the right with specific formatting. Additionally, the program prints the decimal form of the addition and checks for overflow if the result exceeds 64 bits.

Solution Overview:

1. `addHex` Function:

- Converts the input hexadecimal strings (`n1` and `n2`) to unsigned long long integers (`num1` and `num2`) using `sscanf`.

- Adds `num1` and `num2` to calculate `result`.

- Stores the result of addition in hexadecimal format in the `sum` string using `sprintf`.

- Converts the `sum` string to uppercase.

2. `printHex` Function:

- Determines the length of the longer string among `n1` and `n2`.

- Prints `n1`, `n2`, and `sum` in a specific format, aligning to the right with appropriate spacing and a line of dashes beneath `n2`.

3. `main` Function:

- Initializes character arrays (`n1`, `n2`, `sum`) to store hexadecimal numbers.

- Enters a loop to read two hexadecimal strings from input.

- Checks if both input strings are "0"; if yes, exits the loop.

- Calls `addHex` to perform addition and store the result in `sum`.

- Converts input and sum strings to unsigned long long integers for comparison and overflow check.

- Calls `printHex` to print the input numbers and their sum in the specified format.

- Prints the decimal addition of the input numbers and the sum.

- Checks for overflow and prints a message if detected.

Part 2:

Hexadecimal Multiplication Program

Objective:

The program aims to multiply pairs of hexadecimal numbers provided as input until both numbers are "0". Each pair is multiplied together, and the result is printed in a specific format along with the multiplication expression.

Program Structure and Functionality:

1. Header Files:

- \`stdio.h\`: Standard input-output functions.

- \`stdlib.h\`: Standard library functions like \`malloc\`, \`free\`, etc.

- \`string.h\`: String manipulation functions.

- \`ctype.h\`: Character handling functions like \`toupper\`.

2. Constants:

- \`MAX_LENGTH\`: Defines the maximum length of hexadecimal digit strings allowed, considering a maximum of 32 bits (each digit represents 4 bits).

3. Function Definitions:

a. `multiplyHex` Function:

- Purpose: Performs the multiplication of two hexadecimal numbers.

- Steps:

- Reads two hexadecimal strings (`n1` and `n2`) as input.

- Converts input strings to `unsigned long long int` for multiplication.

- Calculates the product and stores it in `product` using `sprintf`.

- Converts the product to uppercase characters using `toupper`.

b. `printHex` Function:

- **Purpose:** Prints the multiplication operation and result in a specific format.

- Steps:

- Computes the total length required for alignment (`len`) based on the input strings.

- Prints the first number (`n1`) aligned to the right with a width of `len`.

- Prints the second number (`n2`) aligned below the '*' sign, right-aligned with width `strlen(n2)`.

- Prints a line of dashes equal to the total length of the numbers (`n1 + n2`).

- Displays the product aligned to the right with a width of `len`.

- Prints the multiplication expression: `n1 * n2 = product` below the product.

4. Main Function:

- Reads input hexadecimal strings in a loop until both are "0".

- Calls `multiplyHex` and `printHex` functions to perform multiplication and display results.