

Programming Assignment 4

Part 1: Hexadecimal

Addition

D1262028 Jimmy 李皓鈞

Code Structure

The code is structured into several sections, including function definitions and the main function. Here's a brief overview of each section:

Function Definitions

`printChar(int n, char c)`

This function prints a character `c` repeated `n` times. It is used to format the output by printing spaces and dashes.

`Hex2decimal(char *Hex)`

This function converts a hexadecimal string to its decimal equivalent. It is crucial for displaying the

decimal sum of the two hexadecimal numbers.

Main Function

The main function contains the core logic of the hexadecimal adder. It reads input, determines the lengths of the input strings, performs hexadecimal addition, and prints the result along with additional information.

Hexadecimal Addition Logic

The program follows a systematic approach to add two hexadecimal numbers. It considers various cases based on the nature of the input digits (0-9 or A-F). The addition is performed from the least significant bit to the most significant bit. The carry is appropriately managed, and the result is formatted for display.

Output Formatting

The output is carefully formatted to align the input numbers, the addition operation, and the result.

The program also checks for overflow and prints a corresponding message if the sum exceeds the allotted 16 bits.

Decimal Conversion

The program includes a function (Hex2decimal) to convert hexadecimal numbers to decimal. This feature enhances the utility of the program by providing both hexadecimal and decimal representations of the sum.

Conclusion

The provided C program successfully implements a hexadecimal adder with robust logic and well-structured code. It handles various cases of input and provides informative output, including both hexadecimal and decimal representations of the

sum. The inclusion of overflow checking adds an extra layer of functionality to the program.

Programming Assignment 4

Part 2: Hexadecimal Multiplication

D1262028 Jimmy 李皓鈞

Introduction

This report provides an overview and analysis of the hexadecimal multiplier implemented in the C programming language. The program takes two hexadecimal numbers as input, performs multiplication, and outputs the product in both hexadecimal and decimal formats. The report discusses the overall structure of the code, key functions, and the design of the hexadecimal multiplier.

Code Structure

The code consists of variable declarations, function definitions, and the main function. Below is a brief overview of each section:

Hex2decimal(char *Hex)

This function converts a hexadecimal string to its decimal equivalent. It is essential for displaying the decimal product of the two hexadecimal numbers.

Main Function

The main function contains the core logic of the hexadecimal multiplier. It reads input, initializes variables, performs hexadecimal multiplication, and prints the result along with additional information.

Hexadecimal Multiplication Logic

The multiplication process involves iterating through each pair of digits in the input numbers (n1

and n^2), calculating partial products, and accumulating the results in the p array. The carry is managed, and the final product is then converted back to hexadecimal and formatted for display.

Output Formatting

The output is carefully formatted to align the input numbers, the multiplication operation, and the result. The program also checks for carry in the first bit and adjusts the output accordingly. The result is displayed in both hexadecimal and decimal formats.

Decimal Conversion

The program includes a function (Hex2decimal) to convert hexadecimal numbers to decimal. This enhances the utility of the program by providing both hexadecimal and decimal representations of the product.

Conclusion

The provided C program successfully implements a hexadecimal multiplier with robust logic and well-structured code. It handles various cases of input and provides informative output, including both hexadecimal and decimal representations of the product. The inclusion of decimal conversion adds versatility to the program. Overall, the hexadecimal multiplier is a useful tool for multiplying hexadecimal numbers efficiently.