

Hexadecimal Addition Function (hexAddition):

The function takes two hexadecimal numbers (n1 and n2) as input and computes their sum.

It pads the shorter number with zeros to make both numbers of equal length.

It iterates through the digits from the least significant to the most significant, performing addition and handling carry.

The result is stored in the sum array.

Hexadecimal to Decimal Conversion Function (hexStringToDecimal):

This function converts a hexadecimal string to its decimal equivalent using the formula $result = result * 16 + decimalValue$.

Printing Function (printChar):

The function prints a character (c) repeated n times.

Main Function:

The main function initializes arrays n1, n2, and sum to store input hexadecimal numbers and their sum.

It reads pairs of hexadecimal numbers until both are "0."

It calculates the sum using the hexAddition function and prints the input numbers, the sum, and the decimal equivalents.

It checks for overflow by comparing the length of the sum with 16 (assuming 16 characters can be used to represent the result).

Input/Output Handling:

The program uses scanf to read input hexadecimal numbers.

It prints the input numbers in a formatted way, aligning them for clarity.

It prints a line of dashes to separate the input numbers from the result.

The result (sum) is printed in both hexadecimal and decimal formats.

Overflow is checked, and a message is printed if the result exceeds the assumed maximum length.

Memory Manipulation (memmove and strlen):

memmove is used to shift characters in the arrays to make room for padding zeros.

strlen is used to determine the length of strings for padding and formatting purposes.

Looping and Conditional Statements:

The program uses a while loop for continuous input until both numbers are "0."
Various loops and conditional statements are used for iterating through digits and performing calculations.

Variable Declarations:

The program declares variables to store lengths (len1, len2, maxLen, l1, l2, lsum), loop variables (i), and input numbers copies (n11 and n21).

Hexadecimal to Decimal Conversion (hexToDecimal):

This function takes a hexadecimal string as input and converts it to its decimal equivalent.

It iterates through each character of the hexadecimal string, updating the decimal value accordingly.

Decimal to Hexadecimal Conversion (decimalToHex):

This function takes a decimal number as input and converts it to its hexadecimal representation.

It repeatedly divides the decimal number by 16, storing the remainders as hexadecimal digits.

The resulting hexadecimal string is then reversed to get the correct order.

Printing Function (printChar):

Similar to the previous program, this function prints a character (c) repeated n times.

Main Function:

The main function initializes arrays n1, n2, and product to store input hexadecimal numbers and their product.

It uses a loop to continuously read pairs of hexadecimal numbers until both are "0."

It calculates the product using the hexToDecimal and decimalToHex functions and prints the input numbers, the product, and the decimal product.

It uses printChar to align the output for clarity.

Memory Manipulation (strlen):

The program uses strlen to determine the length of strings for formatting purposes.

Variable Declarations:

The program declares variables to store lengths (l1, l2, lmul), loop variables (index), and input numbers and result (n1, n2, dec1, dec2, result, product).

Input/Output Handling:

The program uses scanf to read input hexadecimal numbers.

It prints the input numbers in a formatted way, aligning them for clarity.

It prints a line of dashes to separate the input numbers from the result.

The result (product) is printed in both hexadecimal and decimal formats.

Looping and Conditional Statements:

The program uses a while loop for continuous input until both numbers are "0."
Various loops are used inside the conversion functions and for formatting output.