38. Factors Using recursion

#include <stdio.h>

void factors(int n, int i, int \*list, int \*size) {

"""

Generates a list of all factors of a given number `n` using recursion.

Args:

n: The number to find factors for.

i: The current divisor to check (starts from 1).

list: A pointer to the array to store the factors (needs pre-allocation).

size: A pointer to an integer to keep track of the number of factors found.

"""

if (i > n) {

return; // Base case: If the current divisor is greater than `n`, stop.

}

if (n % i == 0) { // Check if `i` is a factor of `n`

list[\*size] = i; // Add `i` to the list

(\*size)++;

factors(n, i, list, size); // Recursively find factors for `n/i`

}

factors(n, i + 1, list, size); // Continue checking with the next divisor

}

int main() {

int n, list[100], size = 0; // Assuming a maximum of 100 factors

printf("Enter a positive integer: ");

scanf("%d", &n);

if (n <= 0) {

printf("Invalid input: Please enter a positive integer.\n");

return 1;

}

factors(n, 1, list, &size);

if (size > 0) {

printf("Factors of %d: ", n);

for (int i = 0; i < size; i++) {

printf("%d ", list[i]);

}

printf("\n");

} else {

printf("%d has no factors other than 1 and itself.\n", n);

}

return 0;

}