Data Structures and Algorithms

Lab 02

Instructions:

- Work on this lab individually. Discussion is not allowed.
- Evaluation of tasks will be conducted in lab.
- Text Files are attached for input.
- Anyone caught being indulged in the act of plagiarism would be awarded an "F" grade in this lab.

Task 1: (10 Marks)

Write a program that takes a binary number (int with only 0's and 1' as digits) as input and displays number of zeros and ones in it. For this purpose, implement two ***recursive functions***:

- countZeros(…)
- countOnes(…)

Task 2: (10 Marks)

Write a ***recursive*** function *B (n, k)*, to compute binomial coefficient, where

- B (n, k) = B (n-1, k-1) + B (n-1, k) for 1 <= k < n
- B (j, 0) = B (j, j) = 1 for j > 0

Find binomial coefficients of the given numbers in the file "Binomial.txt"

Task 3: (10 Marks)

Write a program to Compute GCD of two numbers. Find GCD of numbers given in "GCD.txt"

Description:

What is the greatest common divisor of 54 and 24?

The number 54 can be expressed as a product of two integers in several different ways:

$$54 \times 1 = 27 \times 2 = 18 \times 3 = 9 \times 6.$$

Thus the **divisors of 54** are: $1, 2, 3, 6, 9, 18, 27, 54.$

Similarly, **the divisors of 24** are: $1, 2, 3, 4, 6, 8, 12, 24.$

The numbers that these two lists share in common are the **common divisors** of 54 and 24:

$$1, 2, 3, 6.$$

The greatest of these is 6. That is, the **greatest common divisor** of 54 and 24. One writes:

$$\gcd(54, 24) = 6.$$

Formula:

gcd (a, 0) = a

gcd (a, b) = gcd (b, a mod b)

Instructor: Umm-e-Ammarah

If the arguments are both greater than zero, then the algorithm can be written in more elementary terms as follows

- gcd (a, a) = a
- gcd (a, b) = (a-b, b) if a>b
- gcd (a, b) = (a, b-a) if b>a

Task 4: (10 Marks)

Write a program in C to find the sum of digits of a number using recursion.

Example:

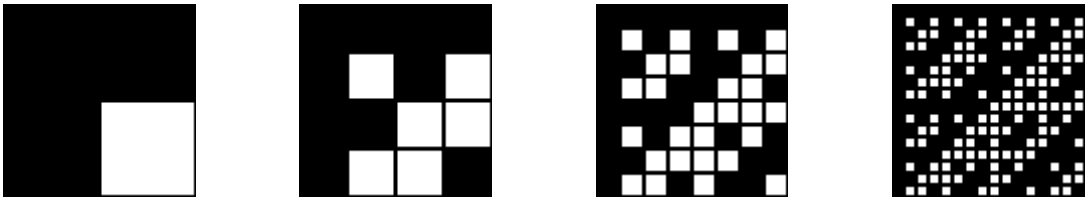
Input any number to find sum of digits: 45

Expected Output:

```
The Sum of digits of 45 = 9
```

Task 5: (20 Marks)

Write a recursive function **Hadamard(int n)** that takes a argument n and plots an N-by-N Hadamard pattern where N = 2n. A 1-by-1 Hadamard pattern is a single black square. In general, a 2N-by-2N Hadamard pattern is obtained by aligning 4 copies of the N-by-N pattern in the form of a 2-by-2 grid, and then inverting the colors of all the squares in the lower right N-by-N copy. The N-by-N Hadamard H(N) matrix is a boolean matrix with the remarkable property that any two rows differ in exactly N/2 bits. This property makes it useful for designing error-correcting codes. Here are the first few Hadamard matrices.



Task 6: (20 Marks)

Using dynamic linear 1D array to hold data, create a class square matrix of order N. You have to create a parameterized constructor, BIG THREE, showMatrix, inputMatrix, and transposeMatrix functions only. Every function will outputs a message on entry and exit. E.g., “entered in transpose function”, “exiting destructor”, etc.

Demonstrate the working of functions in main logic.