# Data Structure and Algorithms

## Problem Set 2: Functions and general problems

**Date of issue:** **Due Date:**

**Problem 1) Palindrome Numbers:** In this problem, you have to read a **positive integer** from the user and pass it into a function name **chk_palindrome** check if it is a **palindrome** or not function will return 1 if a number is palindrome, 0 else, print yes on the return of 1 and no on the return of 0.

A number is a palindrome if the reverse of the number is equal to the number itself.

| | **Input** | **Output** |
|---|---|---|
| Test Case 1 | 34 | no |
| Test Case 2 | 131 | yes |
| Test Case 3 | 1234321 | yes |

**Problem 2) Function to return quotient:** Write a function called **divide** that takes two **non-negative** integers: a and b and returns the quotient of a divided by b, if b is a factor of a, else it returns -1.

| | **Input** | **Output** |
|---|---|---|
| Test Case 1 | 4<br>2 | 2 |
| Test Case 2 | 2<br>4 | -1 |
| Test Case 3 | 2<br>3 | -1 |

**Problem 3) Choose K objects from N distinct object:** Write a C program that calculates the number of ways to choose k objects from n distinct objects. 'k' and 'n' both are integers.

Input Format:
First line contains the value of n, where 0<=n<=10

Second line contains the value of k, where k>=0

Output Format:
One line containing the number of ways to chose the objects

| | **Input** | **Output** |
|---|---|---|
| Test Case 1 | 4<br>2 | 6 |
| Test Case 2 | 2<br>1 | 2 |
| Test Case 3 | 5<br>2 | 10 |

**Problem 4) Missing integer problem:** You are given a sequence of n-1 distinct positive integers, all of which are less than or equal to a integer 'n'. You have to find the integer that is missing from the range [1,2,...,n]. Solve the question without using arrays.

Input Format:
One line containing the integer 'n' where 2<=n<=10,000
First line is followed by a sequence of 'n-1' distinct positive integers. Note that the sequence may not be in any particular order.

Output Format:
One line containing the missing number

| | **Input** | **Output** |
|---|---|---|
| Test Case 1 | 3<br>1 2 | 3 |
| Test Case 2 | 4<br>1 3 4 | 2 |
| Test Case 3 | 4<br>2 1 4 | |

**Problem 5) Counting factorial number:** A number $F$ is a factorial number if there exists some integer $I>=0$ such that $F=I!$ (That is, F is factorial of I). Examples of factorial numbers are:

$1, 2, 6, 24, 120\ldots.$

Write a program that takes as input two **long** integers $n$ and $m$ where $0 \leq n \leq m$. The program prints an integer $k$, where $k$ is the count of factorial numbers in the **closed** interval $[n, m]$.

**Examples**:
Input: 0 1
Output: 1
//Reason (Do not print): Only factorial number is 1

Input: 12 122
Output: 2
// Reason: factorial numbers are 24, 120

Input: 2 720
Output: 5
// Factorial numbers are: 2, 6, 24, 120, 720