Summer Bootcamp in Java

Recursion in java

Assignment # 4
Due Date: 06-Aug-24 till 4:00pm
Marks: 100 marks
Note that: Due date and time will be strictly followed.

Do not copy past any content from other students or internet. It will be considered as plagiarism. Show your own effort and follow all the requirements and instructions. Dear Students, please do it seriously, it will really help you to understand recursive functions.

Task 1:

Write a function that takes a number as its parameter. The function is to return the factorial of that numbergiven by user?

Task 2:

A palindrome is a sequence of characters or numbers that looks the same forwards and backwards. For example, "Madam, I'm Adam" is a palindrome because it is spelled the same reading it from front to back as from back to front. The number 12321 is a numerical palindrome. Write a function that takes a string and its length as arguments and recursively determines whether the string is a palindrome:
int ispalindrome(string s, int len).?

Task 3:

Write a recursive function int count_digit(int n, int digit); to count the number of digits in a number n (n > 0) that are equal to a specified digit. For example, if the digit we're searching for were 2 and the number we're searching were 220, the answer would be 2?

Task 4:

We have *n* people in a room, where *n* is an integer greater than or equal to 1. Each person shakes hands once with every other person. What is the total number, *h(n)* , of handshakes? Write a recursive function to solve this problem.

To get you started, if there are only one or two people in the room, then

handshake(1) = 0
handshake(2) = 1

If a third person enters the room, he or she must shake hands with each of the two people already there. This is two handshakes in addition to the number of handshakes that would be made in a room of two people, or a total of three handshakes. If a fourth person enters the room, this is three handshakes in addition to the number of handshakes that would be made in a room of three people handshakes. If you can generalize this to *n* handshakes, you should be able to write the recursive solution.

Task 5:

The greatest common divisor of integers x and y is the largest integer that evenly divides both x and y.Write a recursive function GCD that returns the greatest common divisor of x and y?

Task 6:

Write a recursive function power (base, exponent) that when invoked returns base ^{exponent}. For example,power (3, 4) = 3 * 3 * 3 * 3. Assume that exponent is an integer greater than or equal to 1?

Task 7:

Write the reverse() function recursively. This function takes a string and the length of the string asarguments and returns the same string with its characters in the reverse order?

Task 8:

You are given a positive integer p. Consider an array nums (1-indexed) that consists of the integers in the inclusive range [1, 2p - 1] in their binary representations. You are allowed to do the following operation any number of times:

Choose two elements x and y from nums.
Choose a bit in x and swap it with its corresponding bit in y. Corresponding bit refers to the bit that is in the same position in the other integer.
For example, if x = 1101 and y = 0011, after swapping the 2nd bit from the right, we have x = 1111 and y = 0001.

Find the minimum non-zero product of nums after performing the above operation any number of times. Return this product modulo 109 + 7.

Note: The answer should be the minimum product before the modulo operation is done.

Example 1:

Input: p = 1
Output: 1
Explanation: nums = [1].
There is only one element, so the product equals that element.

Example 2:

Input: p = 2
Output: 6
Explanation: nums = [01, 10, 11].
Any swap would either make the product 0 or stay the same.
Thus, the array product of 1 * 2 * 3 = 6 is already minimized.

Example 3:

Input: p = 3
Output: 1512
Explanation: nums = [001, 010, 011, 100, 101, 110, 111]
- In the first operation we can swap the leftmost bit of the second and fifth elements.
  - The resulting array is [001, 110, 011, 100, 001, 110, 111].
- In the second operation we can swap the middle bit of the third and fourth elements.
  - The resulting array is [001, 110, 001, 110, 001, 110, 111].

The array product is 1 * 6 * 1 * 6 * 1 * 6 * 7 = 1512, which is the minimum possible product.