**User-defined Functions & Recursive Functions**

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **Find the Factor** |
| **Problem Statement:**  Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the pth element of the list, sorted ascending. If there is no pth element, return 0.  Example n = 20  p = 3  The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if p =  3, then 4 is returned. If p > 6, 0 would be returned.  Function Description  Complete the function pthFactor in the editor below.  pthFactor has the following parameter(s):  int n: the integer whose factors are to be found int p: the index of the factor to be returned  Returns:  int: the long integer value of the pth integer factor of n or, if there is no factor at that index, then 0 is returned  Constraints 1 ≤ n ≤ 1015  1 ≤ p ≤ 109  Input Format for Custom Testing  Input from stdin will be processed as follows and passed to the function. The first line contains an integer n, the number to factor.  The second line contains an integer p, the 1-based index of the factor to return.  Sample Input  STDIN Function    10 → n = 10  3 → p = 3  Sample Output 5  Explanation  Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3rd factor, 5, as the answer. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **Prime or Not?** |
| **Problem Statement:**  Given an integer, if the number is prime, return 1. Otherwise return its smallest divisor greater than 1.  Example n = 24  The number 24 is not prime: its divisors are [1, 2, 3, 4, 6, 8, 12, 24]. The smallest divisor  greater than 1 is 2.  Function Description  Complete the function isPrime in the editor below.  isPrime has the following parameter(s): long n: a long integer to test  Returns  int: if the number is prime, return 1; otherwise returns the smallest divisor greater than 1  Constraints 2 ≤ n ≤ 1012  Input Format for Custom Testing  Input from stdin will be processed as follows and passed to the function. The only line of input contains the long integer to analyze, n.  Sample Input  STDIN Function    2 → n = 2  Sample Output 1  Explanation  As 2 is a prime number, the function returns 1. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **4th Bit** |
| **Problem Statement:**  A binary number is a combination of 1s and 0s. Its nth least significant digit is the nth digit starting from the right starting with 1. Given a decimal number, convert it to binary and determine the value of the the 4th least significant digit.  Example number = 23   * Convert the decimal number 23 to binary number: 2310 = 24 + 22 + 21 + 20 =   (10111)2.   * The value of the 4th index from the right in the binary representation is 0.   Function Description  Complete the function fourthBit in the editor below.  fourthBit has the following parameter(s): int number: a decimal integer  Returns:  int: an integer 0 or 1 matching the 4th least significant digit in the binary representation of number.  Constraints  0 ≤ number < 231  Input Format for Custom Testing  Input from stdin will be processed as follows and passed to the function. The only line contains an integer, number.  Sample Input  STDIN Function    32 → number = 32  Sample Output 0  Explanation   * Convert the decimal number 32 to binary number: 3210 = (100000)2. * The value of the 4th index from the right in the binary representation is 0. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **The Power Sum** |
| **Problem Statement:**  Find the number of ways that a given integer, X, can be expressed as the sum of the Nth powers of unique, natural numbers.  For example, if X = 13 and N = 2, we have to find all combinations of unique squares adding up to 13. The only solution is 22 + 32.  Function Description  Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations.  powerSum has the following parameter(s): X: the integer to sum to  N: the integer power to raise numbers to  Input Format  The first line contains an integer X. The second line contains an integer N.  Constraints  1 ≤ X ≤ 1000  2 ≤ N ≤ 10  Output Format  Output a single integer, the number of possible combinations calculated.  Sample Input 10  2  Sample Output 1  Explanation  If X = 10 and N = 2, we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers.  10 = 12 + 32  This is the only way in which 10 can be expressed as the sum of unique squares. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **Hack the Money** |
| **Problem Statement:**  You are a bank account hacker. Initially you have 1 rupee in your account, and you want exactly N rupees in your account. You wrote two hacks, first hack can multiply the amount of money you own by 10, while the second can multiply it by 20. These hacks can be used any number of time. Can you achieve the desired amount N using these hacks.  Constraints:  1<=T<=100  1<=N<=10^12  Input   * The test case contains a single integer N.   Output  For each test case, print a single line containing the string "1" if you can make exactly N rupees or "0" otherwise.  SAMPLE INPUT 1  SAMPLE OUTPUT 1  SAMPLE INPUT 2  SAMPLE OUTPUT 0 | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **Number of Divisors** |
| **Problem Statement:**  You are given two numbers n and k. For each number in the interval [1, n], your task is to calculate its largest divisor that is not divisible by k. Print the sum of these divisors.  Note: k is a prime number. Input format   * Each test case consists of one line containing two integers n and k.   Output format  The output must contain the answer for each test case on a different line. Each answer consists of a single integer.  Constraints  1 ≤ n ≤ 1000000000  2 ≤ k ≤ 1000000000  SAMPLE INPUT 10 3  SAMPLE OUTPUT 41  SAMPLE INPUT 10 2  SAMPLE OUTPUT 36  Explanation  In the first test case, f (x) from 1 to 10 is [1, 2, 1, 4, 5, 2, 7, 8, 1, 10], sum of which is 41.  In the second test case, f (x) from 1 to 10 is [1, 1, 3, 1, 5, 3, 7, 1, 9, 5]. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |
| --- |
| **Program:** |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

|  |  |
| --- | --- |
| **Ex. No. :** | **Date :** |
|  | **Recursive Digit Sum** |
| **Problem Statement:**  We define super digit of an integer x using the following rules: Given an integer, we need to find the super digit of the integer.   * If x has only 1 digit, then its super digit is x. * Otherwise, the super digit of x is equal to the super digit of the sum of the digits of x. For example, the super digit of 9875 will be calculated as:   super\_digit(9875) 9+8+7+5 = 29, super\_digit(29) 2 + 9 = 11, super\_digit(11) 1 + 1 = 2  super\_digit(2) = 2  You are given two numbers n and k. The number p is created by concatenating the string n k times. Continuing the above example where n = 9875, assume your value k = 4. Your initial p = 9875 9875 9875 9875 (spaces added for clarity).  superDigit(9875987598759875)5+7+8+9+5+7+8+9+5+7+8+9+5+7+8+9 = 116  superDigit(116)1+1+6 = 8, superDigit(8)=8  All of the digits of p sum to 116. The digits of 116 sum to 8. 8 is only one digit, so it's the super digit.  **Function Description**  Complete the function superDigit in the editor below. It must return the calculated super digit as an integer.  **superDigit has the following parameter(s):** n: a string representation of an integer, k: an integer, the times to concatenate n to make p  **Input Format**  The first line contains two space separated integers, n and k.  **Constraints**: 1 ≤ n ≤ 10100000, 1 ≤ k ≤ 105  **Output Format**  Return the super digit of ***p***, where ***p*** is created as described above.  **Sample Input**  148 3  **Sample Output**  3  **Explanation**  Here ***n = 148*** and ***k = 3***, so ***p = 148148148***. super\_digit(P)= super\_digit(148148148)  = super\_digit(1+4+8+1+4+8+1+4+8)  = super\_digit(39)  = super\_digit(3+9)  = super\_digit(12)  = super\_digit(1+2)  = super\_digit(3)  = 3. | |

Department of Computer Science and Engineering, Rajalakshmi Engineering College

### Program:

Department of Computer Science and Engineering, Rajalakshmi Engineering College

142