**QUESTION**

Design a class ArmNum to check if a given number is an Armstrong number or not. [A number is said to be Armstrong if sum of its digits raised to the power of length of the number is equal to the number Example: 371 = 33 + 73 + 13

1634 = 14 + 64 + 34 + 44

54748 = 55 + 45 + 75 + 45 + 85

Thus, 371, 1634 and 54748 are all examples of Armstrong numbers.

Some of the members of the class are given below:

  Class name: ArmNum Data

members/instance variables:

n: to store the number

l: to store the length of the number

Methods/Member functions:

ArmNum (int nn): parameterized constructor to initialize the data member n = nn .

int sum\_pow(int i): returns the sum of each digit raised to the power of the length of the number using recursive technique eg., 34 will return 32 + 42 (as the length of the number is 2)  .

void isArmstrong(): checks whether the given number is an Armstrong number by invoking the function sum\_pow () and displays the result with an appropriate  message.

Specify the class ArmNum giving details of the constructor( ), int sum\_pow(int) and void isArmstrong( ). Define a main() function to create an object and call the functions

**Algorithm**

1. Start

2. Define a class `ArmNum` with the following instance variables:

- `n`: an integer to store the number.

- `l`: an integer to store the length of the number.

3. Define a constructor for the class `ArmNum`:

- Initialize `n` with the parameter `nn`.

- Calculate the length of the number `l` using `String.valueOf(nn).length()`.

4. Define a recursive method `sum\_pow(int i)` for the class `ArmNum`:

- If `i` is 0, return 0.

- Calculate the digit as `i % 10`.

- Return the sum of `Math.pow(digit, l)` and the result of the recursive call `sum\_pow(i / 10)`.

5. Define a method `isArmstrong()` for the class `ArmNum`:

- Call the `sum\_pow(n)` method to get the sum of the digits raised to the power of the length of the number.

- If the sum is equal to `n`, print that `n` is an Armstrong number.

- Otherwise, print that `n` is not an Armstrong number.

6. Define a `main` method for the class `ArmNum`:

- Create a `Scanner` object to read input from the user.

- Prompt the user to enter a number to check if it is an Armstrong number.

- Read the number from the user.

- Create an object of the class `ArmNum` with the entered number.

- Call the `isArmstrong()` method on the created object.

7. End

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| n | int | Stores the number to be checked for being an Armstrong number. |
| l | int | Stores the length of the number n, which is the number of digits in n |
| nn | int | Parameter for the constructor, used to initialize n. |
| i | int | Parameter for the recursive method sum\_pow, used to calculate the sum of the powers of its digits. |
| a | Int | Stores the result of the sum\_pow method, which is the sum of the digits of n each raised to the power of l. |
| x | int | Stores the user input number, which is passed to the constructor to initialize n. |
|  |  |  |

VARIBALE DESCRIPTION TABLE