**Part A**

**1. Write a Java program to find the Armstrong Number.**

import java.util.\*;

public class ArmNum

{

//function to check if the number is Armstrong or not

static boolean isArmstrong(int n)

{

int temp, remainder=0, sum=0;

//assigning n into a temp variable

temp=n;

//to get the number of digits in the number entered by the user

int digits = String.valueOf(n).length();

while(temp>0)

{

//determines the last digit from the number

remainder = temp % 10;

//calculates the power of a number up to digit times and add the resultant to the sum variable

sum +=  (Math.pow(remainder, digits));

//removes the last digit

temp = temp/10;

}

//compares the sum with n

if(n==sum)

//returns if sum and n are equal

return true;

//returns false if sum and n are not equal

else return false;

}

public static void  main(String args[])

{

int num;

Scanner sc= new Scanner(System.in);

System.out.print("Enter the number: ");

//reads the limit from the user

num=sc.nextInt();

if(isArmstrong(num))

{

System.out.print("The number is Armstrong");

}

else

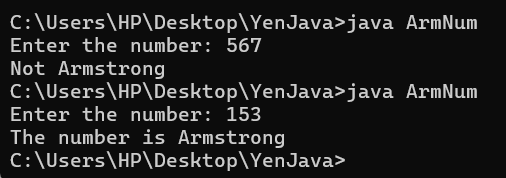
{

System.out.print("Not Armstrong ");

}

}

}



**Explanation**:

Armstrong Number:

An **Armstrong** number is a positive m-digit number that is equal to the sum of the mth powers of their digits. Let’s understand it through an example.

Armstrong Number Example

**1:** 11 = **1**

**2:** 21 = **2**

**3:** 31 = **3**

**153:** 13 + 53 + 33 = 1 + 125+ 27 = **153**

**125:** 13 + 23 + 53 = 1 + 8 + 125 = **134 (Not an Armstrong Number)**

**1634:**14 + 64 + 34 + 44 = 1 + 1296 + 81 + 256 = **1643**

The first few Armstrong numbers between 0 to 999 are **1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407.**

**2. Write a java program that uses both recursive and non-recursive functions to print the Fibonacci Sequence.**

import java.util.Scanner;

class Series

{

int F1, F2=1,F3=0;

short count;

void nonrecursive(short n)

{

count=0;

F1=0;

F2=1;

F3=0;

while (count<n)

{

System.out.println(F1);

F3=F1+F2;

F1=F2;

F2=F3;

count++;

}

}

void recursive(short n)

{

int i=0;

for ( int c = 1 ; c <= n ; c++ )

{

System.out.println(Fib(i));

i++;

}

}

int Fib(int n)

{

if ( n == 0 )

return 0;

else if ( n == 1 )

return 1;

else

return ( Fib(n-1) + Fib(n-2) );

}

}

class Fibonacci

{

public static void main(String args[])

{

System.out.println("Enter the number n to print the fabonicci series : ");

Scanner sc=new Scanner(System.in);

short n=sc.nextShort();

Series ob=new Series();

System.out.println("First " + n + " Fibonacci numbers using recursive function");

ob.recursive(n);

System.out.println("First " + n + " Fibonacci numbers using non-recursive function");

ob.nonrecursive(n);

}

}

