Q1 : Perfect Number using recursion only

Write a java program to find all Perfect numbers between 1 to n using recursion.

Perfect number is a positive integer which is equal to the sum of its proper positive divisors.

For example: 6 is the first perfect number

Proper divisors of 6 are 1, 2, 3

Sum of its proper divisors = 1 + 2 + 3 = 6.

Hence 6 is a perfect number.

**CODE:**

import java.io.\*;

import java.util.\*;

class Solution {

public ArrayList<Integer> perfect(int n) {

// your solutions goes here..

if(n < 6){

ArrayList<Integer> ar = new ArrayList<>();

return ar;

}

ArrayList<Integer> res = perfect(n-1);

int sum =0;

for(int i=1; i<=n/2;i++){

if(n % i == 0){

sum+=i;

}

}

if(sum == n){

res.add(n);

}

return res;

}

}

public class PerfectNumber {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

// Reading N and K

String str = bufferedReader.readLine().trim();

int n = Integer.parseInt(str);

Solution solution = new Solution();

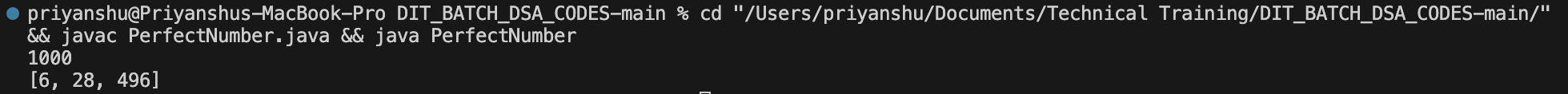
ArrayList<Integer> result = solution.perfect(n);

System.out.println(result);

}

}

**OUTPUT:**

****

Q2 : Convert Octal to Hexadecimal number system using recursion only

Example

Input octal number: 175

Hexadecimal number: 7D

Octal number system

Octal number system is a base 8 number system. It uses 8 symbols to represent all its numbers i.e. 01234567

Hexadecimal number system

Hexadecimal number system is a base 16 number system. It uses 16 symbols to represent all its numbers i.e. 0123456789ABCDEF

**CODE:**

import java.io.\*;

class Solution {

public String convert(int n) {

int decimal = O2D(n); //O2D: Octal to Decimal

return D2H(decimal); //D2H: Decimal to Hexadecimal

}

static int O2D(int octal) {

if (octal == 0) return 0;

int digit = octal % 10;

return digit + 8 \* O2D(octal / 10);

}

static String D2H(int decimal) {

if (decimal == 0) return "";

int remainder = decimal % 16;

char hexDigit = remainder < 10 ? (char) (remainder + '0') : (char) (remainder - 10 + 'A'); //if remainder is ledd than 10 that means we can epresent it beyween 0-9 but

// if incae it's gretaer than or equal to 10 then we have to add 'A' and subtract it with 10 to get the accirate value of hexDigit

return D2H(decimal / 16) + hexDigit;

}

}

public class OctalToHex {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

// Reading N and K

String str = bufferedReader.readLine().trim();

int n = Integer.parseInt(str);

Solution solution = new Solution();

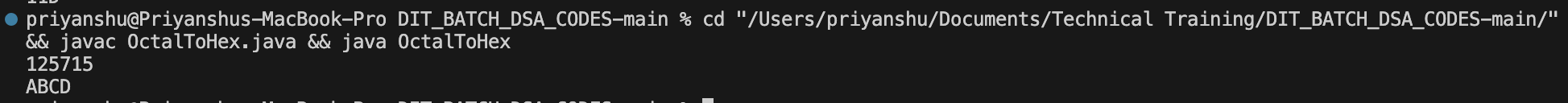
String result = solution.convert(n);

System.out.println(result);

}

}

**OUTPUT:**

****