

```

import java.util.ArrayList;
import java.util.Scanner;

public class EntierSociable {

    public static int somme=0;
    public static int nombrePrecedent;
    public static int ordre=0;
    public static ArrayList <Integer> tableauDiv = new ArrayList<>();

    public static void main(String[] args) {

        Scanner scanner= new Scanner(System.in);
        System.out.println("Entrez la valeur de n (ne pas mettre d'espaces!):");
        double n = scanner.nextInt();
boolean run_the_program = true;
int iterator=0;
int securityNumber=1000000;
        calculDiviseursNombre(n);
        nombrePrecedent=(int)n;
        ordre=0;

        while (somme != n && run_the_program) {
            somme=0;
            iterator++;

            //Pour éviter les boucles infinies
            if(iterator > securityNumber) {
                run_the_program=false;
            }

            for(int k=0; k < tableauDiv.size(); k++) {

                if(tableauDiv.get(k) != nombrePrecedent) {
                    somme+=tableauDiv.get(k);
                }
            }
            nombrePrecedent= somme; System.out.println(somme); tableauDiv.clear();
calculDiviseursNombre(somme); ordre++;
        }

        //Si l'on a pas forcé l'arrêt du programme alors c'est qu'on a bien trouvé l'ordre de
l'entier.
        if(run_the_program != false) {
            System.out.println("Ordre="+ordre);
        }
        else {
            System.out.println("La boucle s'est arrêtée par sécurité, après plus de "+securityNumber+"
calculs, nous n'avons pas pu déterminer l'ordre de l'entier.");
        }
    }

    public static void calculDiviseursNombre(double n) {
        int positionDansLeTableau=0;

        //nous faisons une boucle allant de 2 à la racine carrée de n
        for(int k=1; k < n; k++) {

            if(n%k == 0) {
                tableauDiv.add(positionDansLeTableau, (int) (n/k));
                positionDansLeTableau++;
            }
        }
        tableauDiv.add(positionDansLeTableau, 1);
    }
}

```