**Java Varargs - Simple Addition**

**import** java.io.\*;

**import** java.lang.reflect.\*;

**import** java.util.\*;

**import** java.text.\*;

**import** java.math.\*;

**import** java.util.regex.\*;

**class** Add{

**void** add(**int**...num) {

**int** sum=0;

        String op="";

**for**(**int** i:num) {

            sum=sum+i;

            System.out.print(op+i);

            op="+";

        }

        System.out.println("="+sum);

    }

}

**public** **class** Solution {

**public** **static** **void** main(String[] args) {

**try**{

            BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.in));

**int** n1=Integer.parseInt(br.readLine());

**int** n2=Integer.parseInt(br.readLine());

**int** n3=Integer.parseInt(br.readLine());

**int** n4=Integer.parseInt(br.readLine());

**int** n5=Integer.parseInt(br.readLine());

**int** n6=Integer.parseInt(br.readLine());

            Add ob=**new** Add();

            ob.add(n1,n2);

            ob.add(n1,n2,n3);

            ob.add(n1,n2,n3,n4,n5);

            ob.add(n1,n2,n3,n4,n5,n6);

            Method[] methods=Add.**class**.getDeclaredMethods();

            Set<String> set=**new** HashSet<>();

**boolean** overload=**false**;

**for**(**int** i=0;i<methods.length;i++)

            {

**if**(set.contains(methods[i].getName()))

                {

                    overload=**true**;

**break**;

                }

                set.add(methods[i].getName());

            }

**if**(overload)

            {

**throw** **new** Exception("Overloading not allowed");

            }

            }

**catch**(Exception e)

            {

                e.printStackTrace();

            }

        }

}