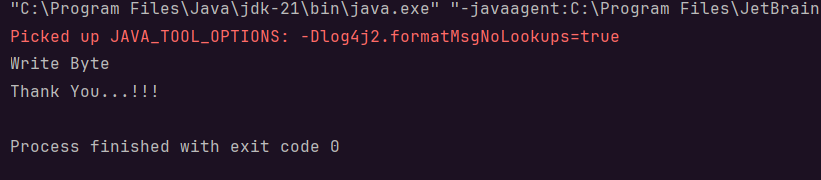
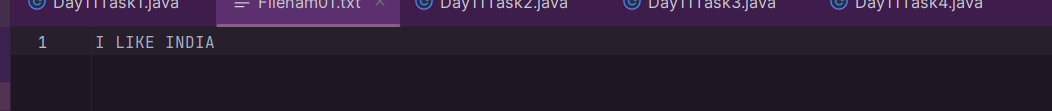
Task1:

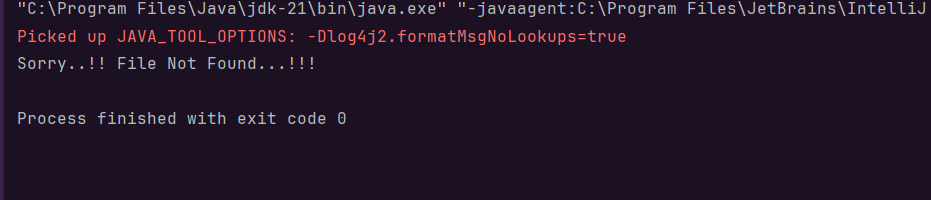
import **java.io.**\*;  
  
public class **Day11Task1**{  
 public static void main(**String** args[])  
 {  
 **File** f1=new File("FileNam01.txt"); //to create new file  
 **FileOutputStream** outfile = null;  
 byte Text[] = {'I',' ', 'L','I','K','E', ' ','I','N','D','I','A'};  
 try  
 {  
 outfile = new FileOutputStream(f1);  
 outfile.write(Text);  
 }  
 catch(**IOException** e)  
 {  
 **System**.*out*.println(e);  
 **System**.*exit*(-1);  
 }  
 **System**.*out*.println("Write Byte");  
 **System**.*out*.println("Thank You...!!!");  
 }  
}





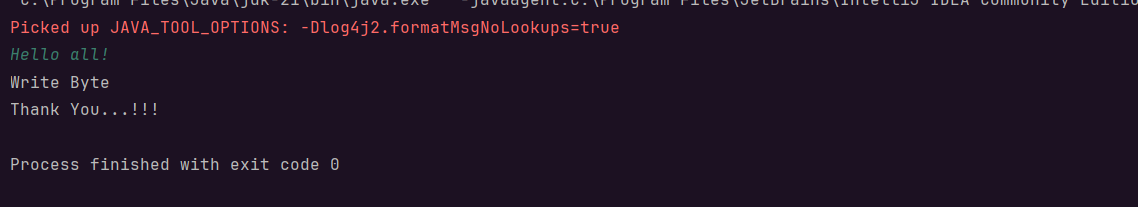
Task2:

import **java.io.**\*;  
public class **Day11Task2**{  
 public static void main(**String** args[])  
 {  
 **FileInputStream** infile = null;  
 int b;  
 try  
 {  
 infile = new FileInputStream("FileName01.txt");  
 while((b = infile.read()) != -1)  
 {  
 **System**.*out*.println((char)b);  
 }  
 infile.close();  
 }  
 catch(**IOException** e)  
 {  
 **System**.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 }  
}



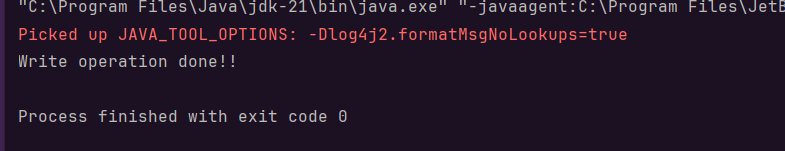
Task3:

import **java.io.**\*;  
import **java.util.**\*;  
public class **Day11Task3**{  
 public static void main(**String** args[]) {  
 **FileOutputStream** outfile = null;  
 //String s=args[0]; // to input string from command line  
 **Scanner** sc=new Scanner(**System**.*in*);  
 **String** s=sc.nextLine();  
 byte b1[] = s.getBytes();  
 try  
 {  
 outfile = new FileOutputStream("FileName02.txt");  
 outfile.write(b1);  
 }  
 catch(**IOException** e)  
 {  
 **System**.*out*.println(e);  
 **System**.*exit*(-1);  
 }  
 **System**.*out*.println("Write Byte");  
 **System**.*out*.println("Thank You...!!!");  
 }  
}



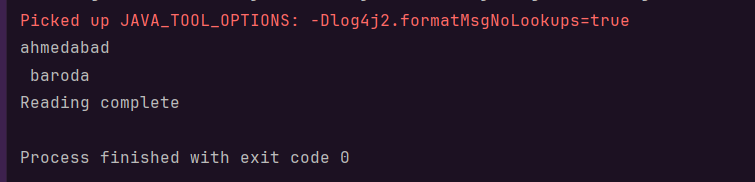
Task4:

import **java.io.**\*;  
class **Day11Task4** {  
public static void main(**String** args[]) {  
 **File** f1=new File("FileName03.txt");  
 **FileWriter** fw = null;  
 try {  
 fw=new FileWriter(f1);  
 fw.write("ahmedabad **\n**");  
 fw.write(" baroda **\n**");  
 fw.close();  
 }  
 catch(**FileNotFoundException** e)  
 {  
 **System**.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 catch(**IOException** e)  
 {  
 **System**.*out*.println(e.getMessage());  
 }  
 **System**.*out*.println("Write operation done!!");  
}  
}



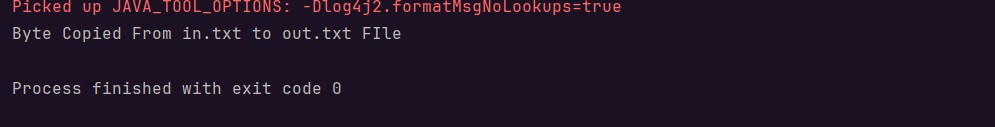
Task5:

import **java.io.**\*;  
class **Day11Task5**{  
public static void main(**String** args[])  
{  
 **FileReader** fr =null;  
 try  
 {  
 fr = new FileReader("FileName03.txt");  
 int ch;  
 while((ch = fr.read()) != -1)  
 {  
 **System**.*out*.print((char)ch);  
 }  
 **System**.*out*.println("Reading complete");  
 fr.close();  
 }  
 catch(**FileNotFoundException** e)  
 {  
 **System**.*out*.println("Sorry..!! File Not Found...!!!");  
 }  
 catch(**IOException** e)  
 {  
 **System**.*out*.println(e.getMessage());  
 }  
}  
}



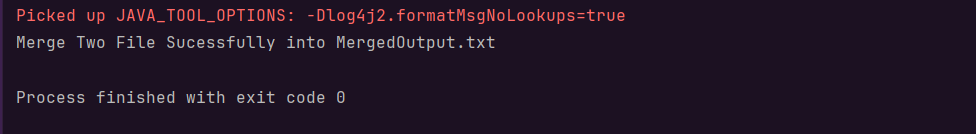
Task6:

import **java.io.**\*;  
  
public class **Day11Task6**{  
  
 public static void main(**String** args[])  
  
 {  
  
 try  
  
 {  
  
 byte b=0;  
  
 **FileInputStream** infile = new FileInputStream("Filename02.txt");  
  
 **FileOutputStream** outfile = new FileOutputStream("FileName03.txt");  
  
 // Initialize byteread here….  
  
 int byteread = infile.read();  
  
 while(byteread != -1)  
  
 {  
  
 b=(byte) byteread;  
  
 outfile.write(b);  
  
 byteread = infile.read();  
  
 }  
  
 **System**.*out*.println("Byte Copied From in.txt to out.txt FIle ");  
  
 }  
  
 catch(**FileNotFoundException** e)  
  
 {  
  
 **System**.*out*.println("Sorry..!! File Not Found...!!!");  
  
 }  
  
 catch(**IOException** e)  
  
 {  
  
 **System**.*out*.println(e.getMessage());  
  
 }  
  
 }  
  
}



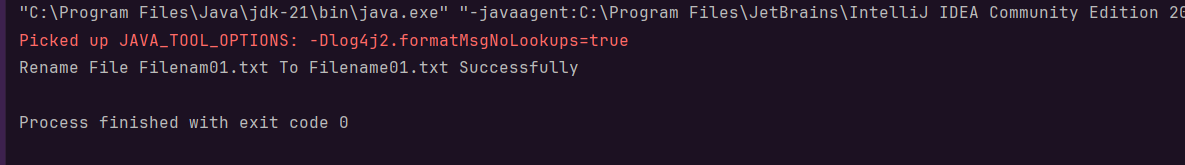
Task7:

import **java.io.**\*;  
  
class **Day11Task7** {  
 public static void main(**String** args[]) {  
 **String** outputFileName = "MergedOutput.txt";  
  
 try (**FileInputStream** file1 = new FileInputStream("Filenam01.txt");  
 **FileInputStream** file2 = new FileInputStream("FileName02.txt");  
 **SequenceInputStream** file3 = new SequenceInputStream(file1, file2);  
 **BufferedInputStream** br1 = new BufferedInputStream(file3);  
 **FileOutputStream** outputFileStream = new FileOutputStream(outputFileName);  
 **BufferedOutputStream** br2 = new BufferedOutputStream(outputFileStream)) {  
  
 int ch;  
 while ((ch = br1.read()) != -1) {  
 br2.write(ch);  
 }  
  
 **System**.*out*.println("Merge Two File Sucessfully into " + outputFileName);  
  
 } catch (**FileNotFoundException** e) {  
 **System**.*out*.println("Sorry..!! One of the input files not found...!!!");  
 **System**.*out*.println("Error: " + e.getMessage());  
 } catch (**IOException** e) {  
 **System**.*out*.println("An I/O error occurred: " + e.getMessage());  
 }  
 }  
}



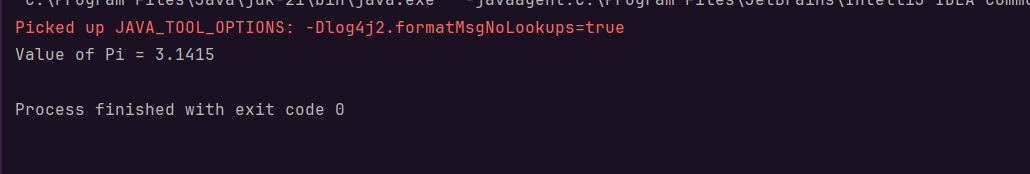
Task8:

import **java.io.File**;  
  
public class **Day11Task8** {  
  
 public static void main(**String**[] args) {  
  
 **File** f1=new File("Filenam01.txt");  
  
 **File** f2= new File("Filename01.txt");  
  
 boolean success = f1.renameTo(f2);  
  
 if (success) {  
  
 **System**.*out*.println("Rename File " +f1+" To "+f2+" Successfully ");  
  
 } else {  
  
 **System**.*out*.println("Failed to rename file.");  
  
 }  
  
  
 }  
  
}



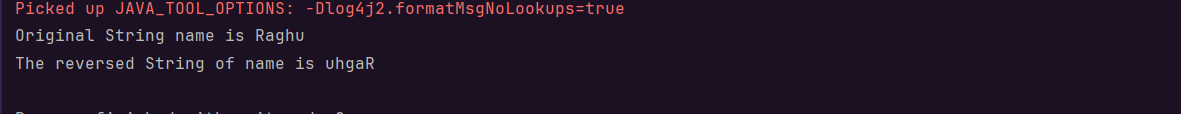
Task9:

import **java.lang.**FunctionalInterface;  
  
// this is functional interface  
@FunctionalInterface  
interface **MyInterface**{  
  
 // abstract method  
 double getPiValue();  
}  
  
public class **Day11Task9** {  
  
 public static void main( **String**[] args ) {  
  
 // declare a reference to MyInterface  
 **MyInterface** ref;  
  
 // lambda expression  
 ref = () -> 3.1415;  
  
 **System**.*out*.println("Value of Pi = " + ref.getPiValue());  
 }  
}



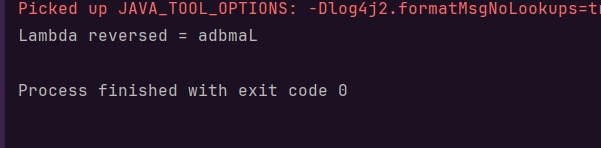
Task10:

public class **Day11Task10** {  
 public static void main(**String**[] args) {  
 **String** name = "Raghu";  
 **String** ReverseString = "";  
 for (int i=name.length()-1; i>=0; i--){  
 ReverseString += name.charAt(i);  
 }  
 **System**.*out*.println("Original String name is "+ name);  
 **System**.*out*.println("The reversed String of name is "+ReverseString);  
 }  
}



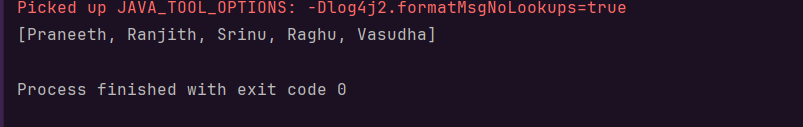
Task11:

@FunctionalInterface  
interface **MyInterface** {  
  
 // abstract method  
 **String** reverse(**String** n);  
}  
  
public class **Day11Task11** {  
  
 public static void main( **String**[] args ) {  
  
 // declare a reference to MyInterface  
 // assign a lambda expression to the reference  
 **MyInterface** ref = (str) -> {  
  
 **String** result = "";  
 for (int i = str.length()-1; i >= 0 ; i--)  
 result += str.charAt(i);  
 return result;  
 };  
  
 // call the method of the interface  
 **System**.*out*.println("Lambda reversed = " + ref.reverse("Lambda"));  
 }  
  
}



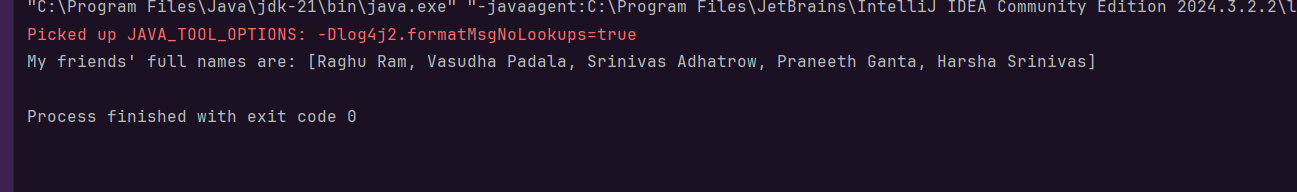
Task12:

import **java.util.ArrayList**;  
  
public class **Day11Task12** {  
 public static void main(**String**[] args) {  
 **ArrayList**<**String**> friends = new ArrayList<>();  
 friends.add("Praneeth");  
 friends.add("Ranjith");  
 friends.add("Srinu");  
 friends.add("Raghu");  
 friends.add("Vasudha");  
 **System**.*out*.println(friends);  
 }  
}



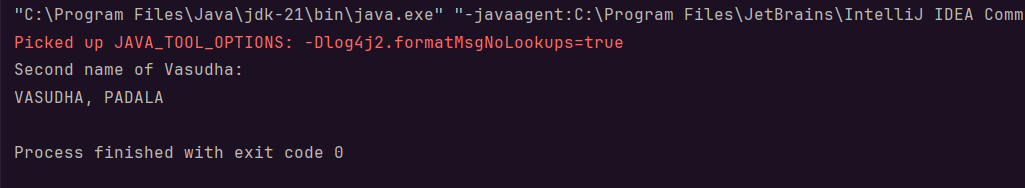
Task13:

import **java.util.LinkedList**; import **java.util.List**;  
  
public class **Day11Task13** {  
  
 public static void main(**String**[] args) {  
  
 **List**<**String**> firstNames = **List**.*of*(  
  
 "Raghu",  
  
 "Vasudha",  
  
 "Srinivas",  
  
 "Praneeth",  
  
 "Harsha"  
  
 );  
  
 **List**<**String**> lastNames = **List**.*of*("Ram", "Padala", "Adhatrow", "Ganta", "Srinivas");  
  
 **List**<**String**> fullNames = new LinkedList<>();  
  
 for (int i = 0; i < firstNames.size(); i++) {  
  
 **String** firstName = firstNames.get(i);  
  
 **String** lastName = lastNames.get(i);  
  
 **String** fullName = firstName + " " + lastName;  
  
 fullNames.add(fullName);  
  
 }  
  
 **System**.*out*.println("My friends' full names are: " + fullNames);  
  
 }  
  
}



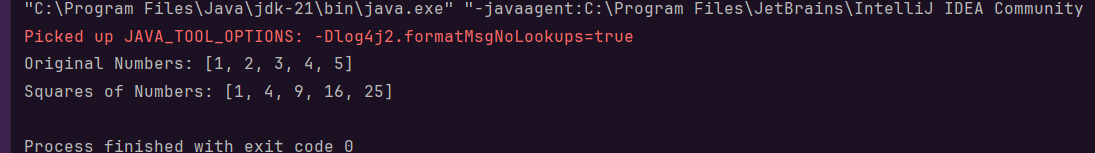
Task14:

import **java.util.ArrayList**;  
import **java.util.List**;  
  
public class **Day11Task14** {  
  
 // create an object of list using ArrayList  
 static **List**<**String**> *Names* = new ArrayList<>();  
  
 // preparing our data  
 public static **List** getNames(){  
  
 // add places and country to the list  
 *Names*.add("Raghu, Ponamgi");  
 *Names*.add("Vasudha, Padala");  
 *Names*.add("Praneeth, Ganta");  
 *Names*.add("Srinivas, Adhatrow");  
  
  
  
 return *Names*;  
 }  
  
 public static void main( **String**[] args ) {  
  
 **List**<**String**> myNames = *getNames*();  
 **System**.*out*.println("Second name of Vasudha:");  
  
  
 myNames.stream()  
 .filter((p) -> p.startsWith("Vasudha"))  
 .map((p) -> p.toUpperCase())  
 .sorted()  
 .forEach((p) -> **System**.*out*.println(p));  
 }  
  
}



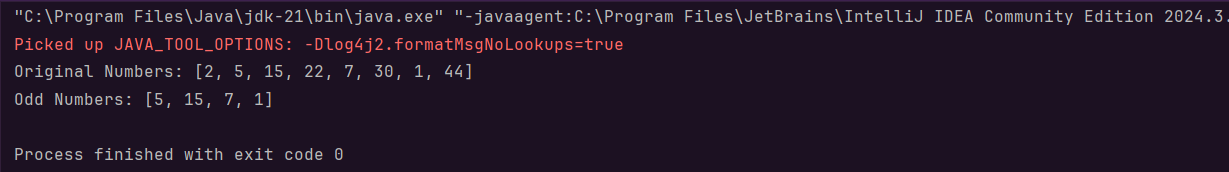
Task15:

import **java.util.ArrayList**;  
import **java.util.List**;  
import **java.util.stream.Collectors**;  
  
 public class **Day11Task15** {  
 public static void main(**String**[] args) {  
 **List**<**Integer**> numbers = new ArrayList<>();  
  
 numbers.add(1);  
 numbers.add(2);  
 numbers.add(3);  
 numbers.add(4);  
 numbers.add(5);  
  
 **System**.*out*.println("Original Numbers: " + numbers);  
  
 **List**<**Integer**> squareOfNums = numbers.stream()  
 .map(num -> num \* num)  
 .collect(**Collectors**.*toList*());  
  
 **System**.*out*.println("Squares of Numbers: " + squareOfNums);  
 }  
}



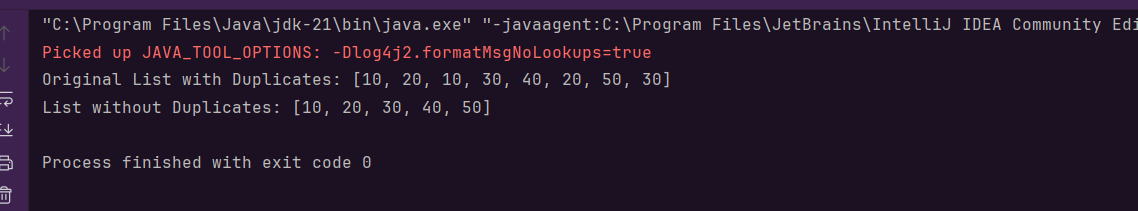
Task16:

import **java.util.ArrayList**;  
import **java.util.List**;  
import **java.util.stream.Collectors**;  
  
public class **Day11Task16** {  
 public static void main(**String**[] args) {  
 **List**<**Integer**> numbers = new ArrayList<>();  
 numbers.add(2);  
 numbers.add(5);  
 numbers.add(15);  
 numbers.add(22);  
 numbers.add(7);  
 numbers.add(30);  
 numbers.add(1);  
 numbers.add(44);  
 **System**.*out*.println("Original Numbers: "+ numbers);  
 **List**<**Integer**> oddNumbers = numbers.stream().filter(num ->num %2 !=0).collect(**Collectors**.*toList*());  
 **System**.*out*.println("Odd Numbers: "+ oddNumbers);  
 }  
}



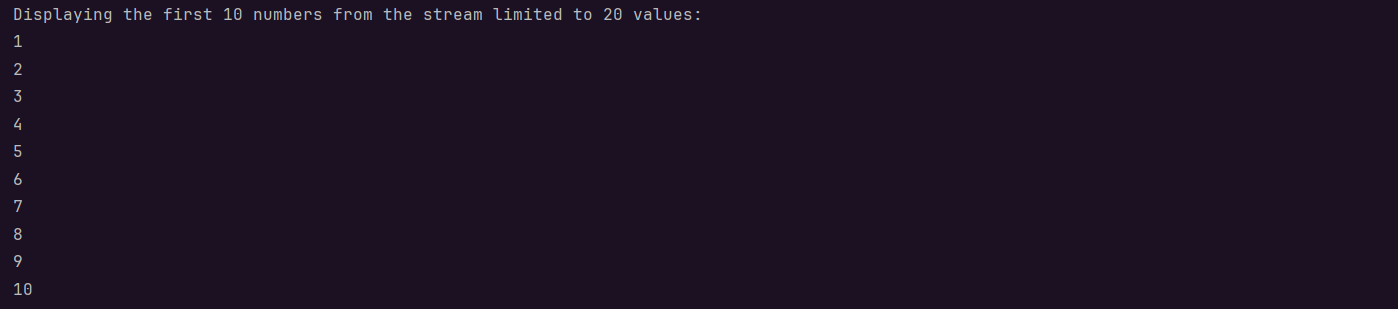
Task17:

import **java.util.ArrayList**;  
import **java.util.List**;  
import **java.util.stream.Collectors**;  
  
public class **Day11Task17** {  
 public static void main(**String**[] args) {  
 **List**<**Integer**> numbersWithDuplicates = new ArrayList<>();  
 numbersWithDuplicates.add(10);  
 numbersWithDuplicates.add(20);  
 numbersWithDuplicates.add(10);  
 numbersWithDuplicates.add(30);  
 numbersWithDuplicates.add(40);  
 numbersWithDuplicates.add(20);  
 numbersWithDuplicates.add(50);  
 numbersWithDuplicates.add(30);  
  
 **System**.*out*.println("Original List with Duplicates: " + numbersWithDuplicates);  
  
 **List**<**Integer**> numbersWithoutDuplicates = numbersWithDuplicates.stream()  
 .distinct()  
 .collect(**Collectors**.*toList*());  
  
 **System**.*out*.println("List without Duplicates: " + numbersWithoutDuplicates);  
 }  
}



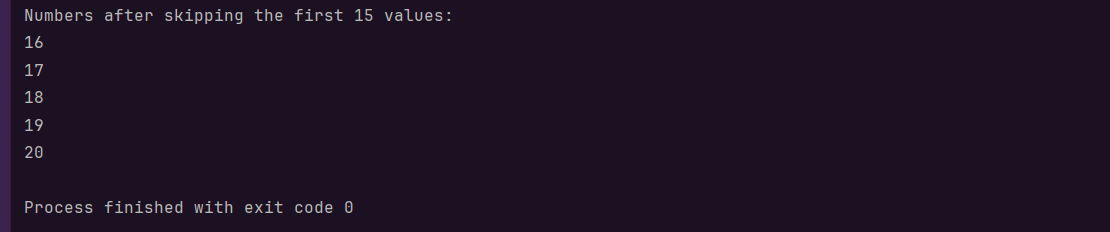
Task18:

import **java.util.stream.Stream**;  
  
public class **Day11Task18** {  
 public static void main(**String**[] args) {  
 **Stream**<**Integer**> nums = **Stream**.*iterate*(1, n -> n + 1)  
 .limit(20);  
  
 **System**.*out*.println("Displaying the first 10 numbers from the stream limited to 20 values:");  
  
 nums.limit(10)  
 .forEach(**System**.*out*::println);  
 }  
}



Task19:

import **java.util.stream.Stream**;  
  
public class **Day11Task19** {  
 public static void main(**String**[] args) {  
 **Stream**<**Integer**> nums = **Stream**.*iterate*(1, n -> n + 1)  
 .limit(20);  
  
 **Stream**<**Integer**> skipNums = nums.skip(15);  
  
 **System**.*out*.println("Numbers after skipping the first 15 values:");  
  
 skipNums.forEach(**System**.*out*::println);  
 }  
}



Task20:

import **java.util.Arrays**;  
import **java.util.List**;  
import **java.util.Optional**;  
  
public class **Day11Task20** {  
 public static void main(**String**[] args) {  
  
 **List**<**Integer**> numbers = **Arrays**.*asList*(1, 2, 3, 4, 5);  
  
 **Optional**<**Integer**> sum = numbers.stream().reduce((x, y) -> x + y);  
 **System**.*out*.println("Sum of all elements: " + sum.orElse(0));  
  
  
 **Optional**<**Integer**> max = numbers.stream().reduce(**Integer**::*max*);  
 **System**.*out*.println("Maximum element: " + max.orElse(0));  
  
 **List**<**String**> strings = **Arrays**.*asList*("Hello", " ", "world", "!");  
 **Optional**<**String**> concatenatedString = strings.stream().reduce((x, y) -> x + y);  
 **System**.*out*.println("Concatenated string: " + concatenatedString.orElse(""));  
 }  
}

