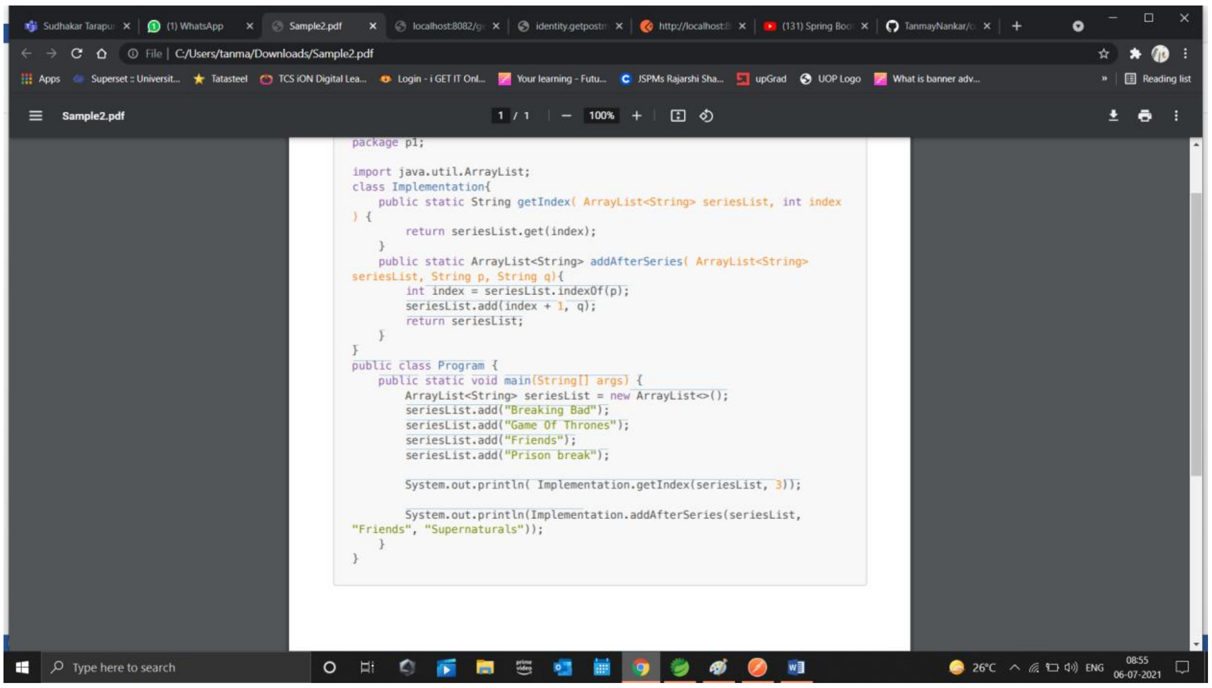
What is an Age

```
package test;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

class Person{
	private String name;
	private int age;
	public Person(String name, int age) {
		this.name = name;
		this.age = age;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public int getAge() {
		return age;
	}
	public void setAge(int age) {
		this.age = age;
	}
}
class StreamImplementation{
	public static int sumAge( List<Person> list ) {
		return list.stream().filter(p->p.getAge() > 50 ).mapToInt(p->p.getAge()).sum();
	}
	public static List<String> printName( List<Person> list ) {
		return list.stream().map( p-> p.getName( ) ).collect(Collectors.toList());
	}
	public static List<Integer> printAge( List<Person> list ) {
		return list.stream().map(p->p.getAge()).collect(Collectors.toList());
	}
}

public class Program {
	public static void main(String[] args) {
		List<Person> list = new ArrayList<Person>( );
		list.add(new Person("Perry", 20));
		list.add(new Person("Ferry", 52));
		list.add(new Person("Katty", 100));
		list.add(new Person("Elly", 14));

		System.out.println(StreamImplementation.sumAge(list));

		System.out.println(StreamImplementation.printName(list));

		System.out.println(StreamImplementation.printAge(list));
	}
}
```

:add elements to end using condition



```
package p1;

import java.util.ArrayList;
class Implementation{
    public static String getIndex( ArrayList<String> seriesList, int index
) {
        return seriesList.get(index);
    }
    public static ArrayList<String> addAfterSeries( ArrayList<String>
seriesList, String p, String q){
        int index = seriesList.indexOf(p);
        seriesList.add(index + 1, q);
        return seriesList;
    }
}
public class Program {
    public static void main(String[] args) {
        ArrayList<String> seriesList = new ArrayList<>();
        seriesList.add("Breaking Bad");
        seriesList.add("Game Of Thrones");
        seriesList.add("Friends");
        seriesList.add("Prison break");

        System.out.println( Implementation.getIndex(seriesList, 3));

        System.out.println(Implementation.addAfterSeries(seriesList,
"Friends", "Supernaturals"));
    }
}
```

List of operation

```
import java.util.*;
class ArrayListOps {

	public List makeArrayListInt(int n){
		List<Integer> arrList = new ArrayList<Integer>();
		for(int i = 0 ; i < n ; i++){
			arrList.add(0);
		}
		return arrList;
	}
	public List reverseList(ArrayList<Integer> arr){
		Collections.reverse(arr);
		return arr;
	}
	public List changeList(ArrayList<Integer> arrL, int m, int n){
		Collections.replaceAll(arrL,m,n);
		return arrL;
	}
}
public class Source{
	public static void main(String[] args) {
	}
}
```

Age

```
import java.util.*;
class Age {
String drinkingAge;
}
class implementation {
public String validateAgeToDrink(Age obj, int personAge) {
try {
 if(personAge<21) {
 throw new IllegalAgeException("Illegal drinking age");
 } else {
 obj.drinkingAge = "legal";
 }
 } catch(IllegalAgeException iae) {
 obj.drinkingAge = "illegal";
 return iae.getMessage();
 }
return obj.drinkingAge;
}
public String validateStringAgeToDrink(Age obj, String personAge) {
try { if(personAge.length()>2 || Integer.parseInt(personAge)<21) {
 throw new IllegalAgeException("Invalid age detail or drinking age");
 } else if(personAge.length()<2 && Integer.parseInt(personAge)>=21) {
 obj.drinkingAge = "legal";
 }
} catch(IllegalAgeException iae) {
 obj.drinkingAge = "illegal";
 return iae.getMessage();
}
 return obj.drinkingAge;
}
}
class IllegalAgeException extends Exception {
IllegalAgeException(String str) {
 super(str);
}
}
public class Source {
public static void main(String args[]) throws Exception {
 }
}
```

School Management System

```
public class Student {
 private String name;
 private float percentage;
 public Student(String name, float percentage) {
  super();
  this.name = name;
  this.percentage = percentage;
 }
 public String getName() {
  return name;
 }
 public void setName(String name) {
  this.name = name;
 }
 public float getPercentage() {
  return percentage;
 }
 public void setPercentage(float percentage) {
  this.percentage = percentage;
 }

 public static void main(String[] args) {
  ArrayList<Student> list = new ArrayList<>();
  list.add(new Student("Steve",(float)56.3));
  list.add(new Student("Bob",(float)67.3));
  list.add(new Student("Alice",(float)98.4));
  list.add(new Student("Mark",(float)40));
  School obj = new School();
  obj.studentList = list;




  obj.sortByName();
  obj.getAvgPercentage();
 }
}

class Sorting implements Comparator<Student>
{

 @Override
 public int compare(Student o1, Student o2) {

  if((o1.getName().compareTo(o2.getName())) > 0)
  {
   return 1;
  }else
  {
   return -1;
  }

 }

}

class School
{
```

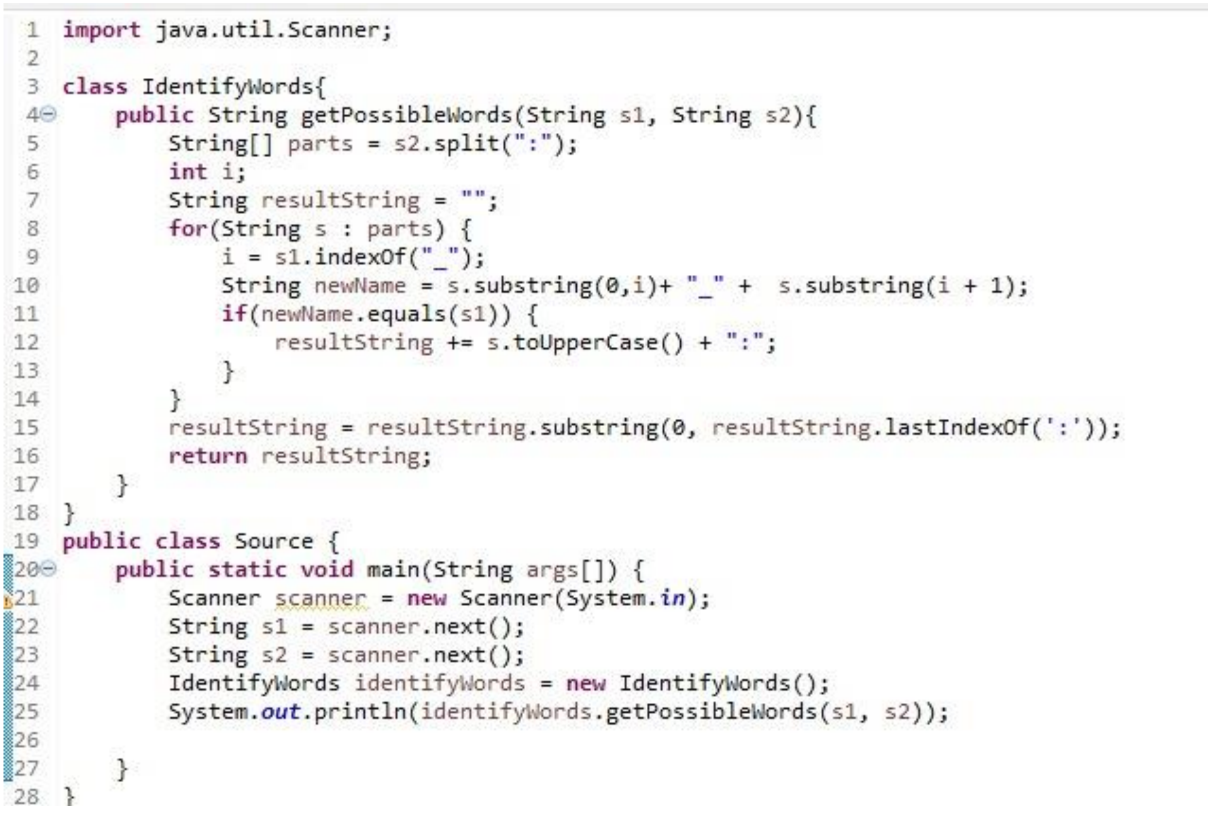
```
ArrayList<Student> studentList = new ArrayList<Student>();

public ArrayList<Student> sortByName ()
{
 Sorting s = new Sorting();
 Collections.sort(studentList,s);
 return studentList;
}

public double getAvgPercentage()
{
 double Avgp = 0;
 for(Student s : studentList)
 {
  Avgp = Avgp + s.getPercentage();
 }

 Avgp = Avgp/studentList.size();

 return Avgp;
}
}
```

Set first and last name

```
public class Employee {

 private String firstName;
 private String lastName;
 private String ssn;

 public Employee()
 {
  firstName = null;
  lastName = null;
  ssn = null;
 }

 public Employee(String firstName, String lastName, String ssn) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.ssn = ssn;
 }

 public String getFirstName() {
  return firstName;
 }

 public void setFirstName(String firstName) {
  this.firstName = firstName;
```

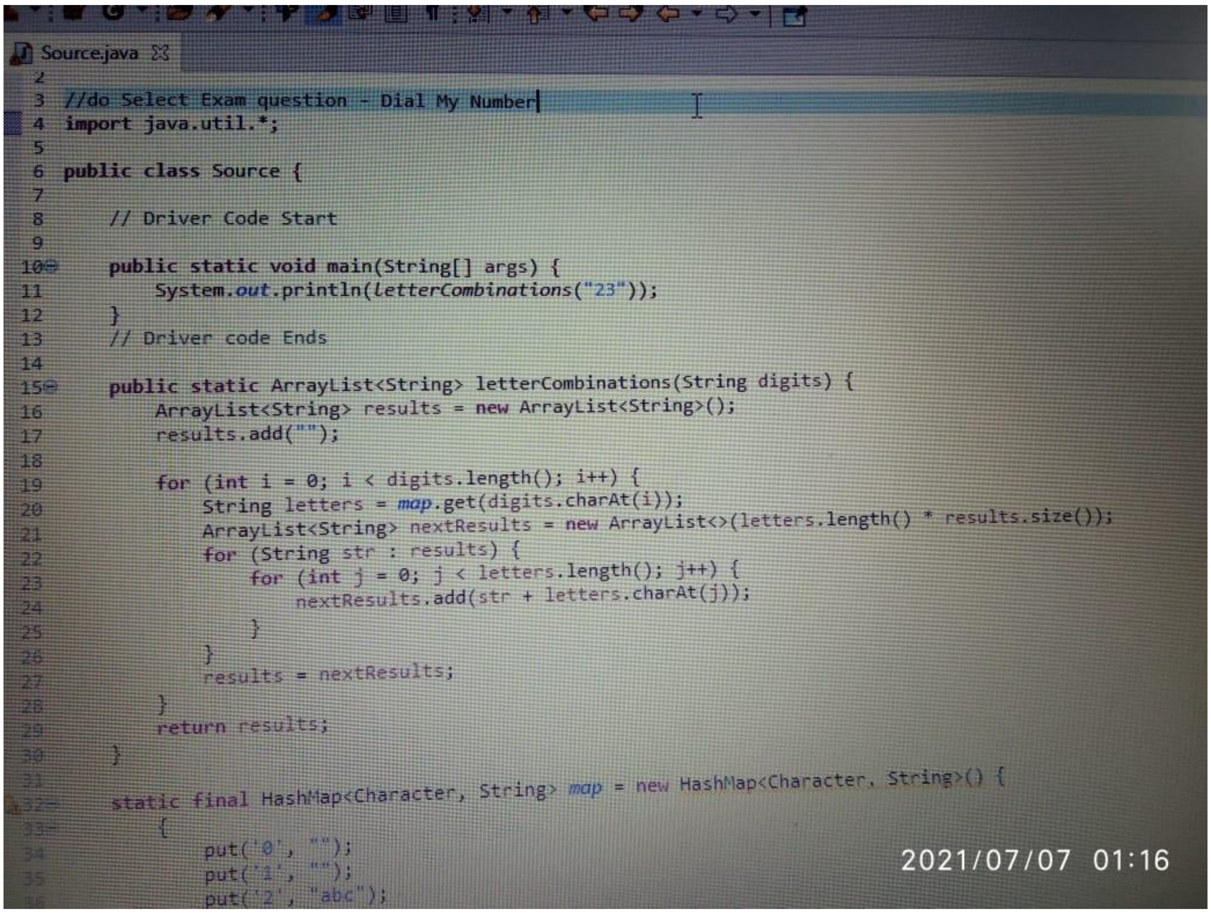
```
}

public String getLastName() {
 return lastName;
}

public void setLastName(String lastName) {
 this.lastName = lastName;
}

public String getSsn() {
 return ssn;
}

public void setSsn(String ssn) {
 this.ssn = ssn;
}

String validateName(String firstName,String lastName) throws Exception
{
 try
 {
  if(firstName == null || lastName == null)
  {
   Exception NullPointerException = new NullPointerException("Entry Missing");
   throw NullPointerException;
  }
  else if(firstName.length() == 0 || lastName.length() == 0)
  {
    Exception StringIndexOutOfBoundsException = new
StringIndexOutOfBoundsException("Index out of bound");

    throw StringIndexOutOfBoundsException;

   }else if(Character.isDigit(firstName.charAt(0)) || Character.isDigit(lastName.charAt(0)))

   {

    Exception IllegalArgumentException = new IllegalArgumentException("First Character is
invalid");

    throw IllegalArgumentException;

   }


   setFirstName(firstName);

   setLastName(lastName);

   return "Valid String";


  }catch(NullPointerException n)

  {

   n.getMessage();

   return "";

  }

  catch(StringIndexOutOfBoundsException s)

  {

   s.getMessage();

   return "";

  }

  catch(IllegalArgumentException i)

  {

   i.getMessage();

   return "";

  }

 } }
```

Sherlock Needs Help

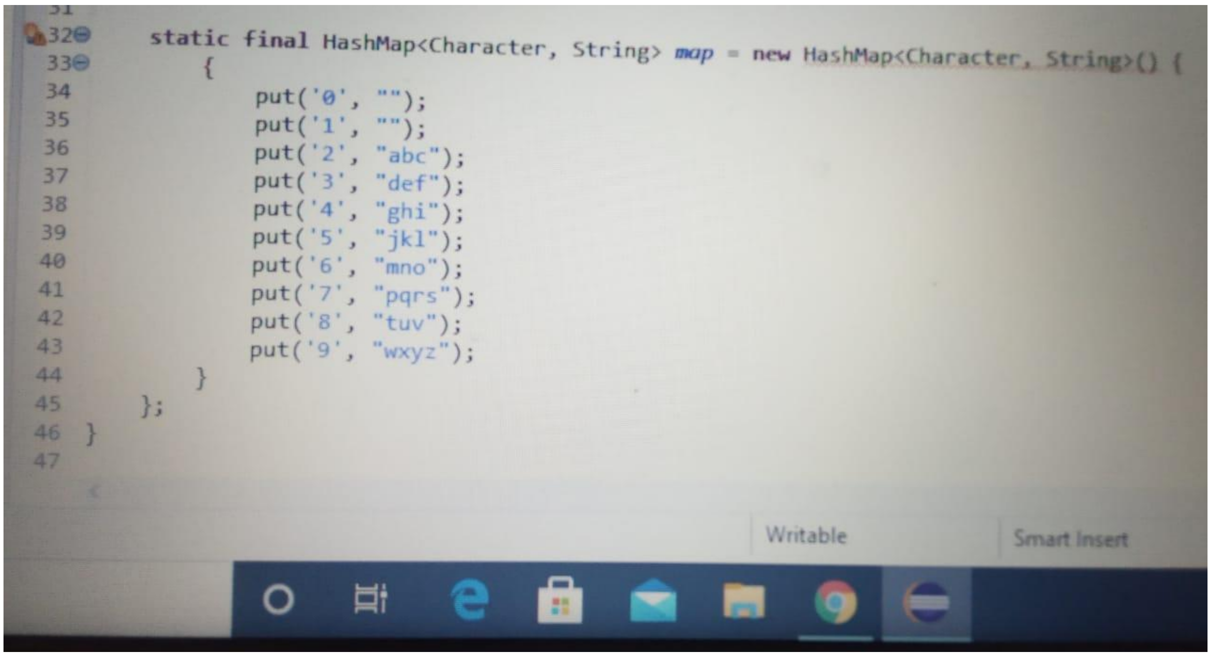


```java
import java.util.Scanner;

class IdentifyWords{
    public String getPossibleWords(String s1, String s2){
        String[] parts = s2.split(":");
        int i;
        String resultString = "";
        for(String s : parts) {
            i = s1.indexOf("_");
            String newName = s.substring(0,i)+ "_" +  s.substring(i + 1);
            if(newName.equals(s1)) {
                resultString += s.toUpperCase() + ":";
            }
        }
        resultString = resultString.substring(0, resultString.lastIndexOf(':'));
        return resultString;
    }
}
public class Source {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        String s1 = scanner.next();
        String s2 = scanner.next();
        IdentifyWords identifyWords = new IdentifyWords();
        System.out.println(identifyWords.getPossibleWords(s1, s2));

    }
}
```

Q-Dial my number



```java

//do Select Exam question - Dial My Number
import java.util.*;

public class Source {

    // Driver Code Start

    public static void main(String[] args) {
        System.out.println(letterCombinations("23"));
    }
    // Driver code Ends

    public static ArrayList<String> letterCombinations(String digits) {
        ArrayList<String> results = new ArrayList<String>();
        results.add("");

        for (int i = 0; i < digits.length(); i++) {
            String letters = map.get(digits.charAt(i));
            ArrayList<String> nextResults = new ArrayList<>(letters.length() * results.size());
            for (String str : results) {
                for (int j = 0; j < letters.length(); j++) {
                    nextResults.add(str + letters.charAt(j));
                }
            }
            results = nextResults;
        }
        return results;
    }

    static final HashMap<Character, String> map = new HashMap<Character, String>() {
        {
            put('0', "");
            put('1', "");
            put('2', "abc");
```



```java
    static final HashMap<Character, String> map = new HashMap<Character, String>() {
        {
            put('0', "");
            put('1', "");
            put('2', "abc");
            put('3', "def");
            put('4', "ghi");
            put('5', "jkl");
            put('6', "mno");
            put('7', "pqrs");
            put('8', "tuv");
            put('9', "wxyz");
        }
    };
}
```

: BMI Calculator

```
import java.util.Scanner;

public class BmiCalculator {

        float getWeight (String str) {
                String [] weightArray=str.split("@");
                String wt = (weightArray[0].replace("-", "."));
                float weight = Float.parseFloat(wt);
                return weight;

        }
        float getHeight(String str) {
                String [] HeightArray= str.split("@");
                String ht =(HeightArray[1].replace("-", "."));
                float height = Float.parseFloat(ht);
                return height;
        }

        public static void main(String[] args) {
                BmiCalculator bmiCal = new BmiCalculator();
                Scanner scanner = new Scanner(System.in);
                System.out.println("Please enter string");
                String weight = scanner.next();
                float wt = bmiCal.getWeight(weight);
                System.out.println(wt);

                float ht = bmiCal.getWeight(weight);
                System.out.println(ht);
        }
}
```