Vellore Institute of Technology

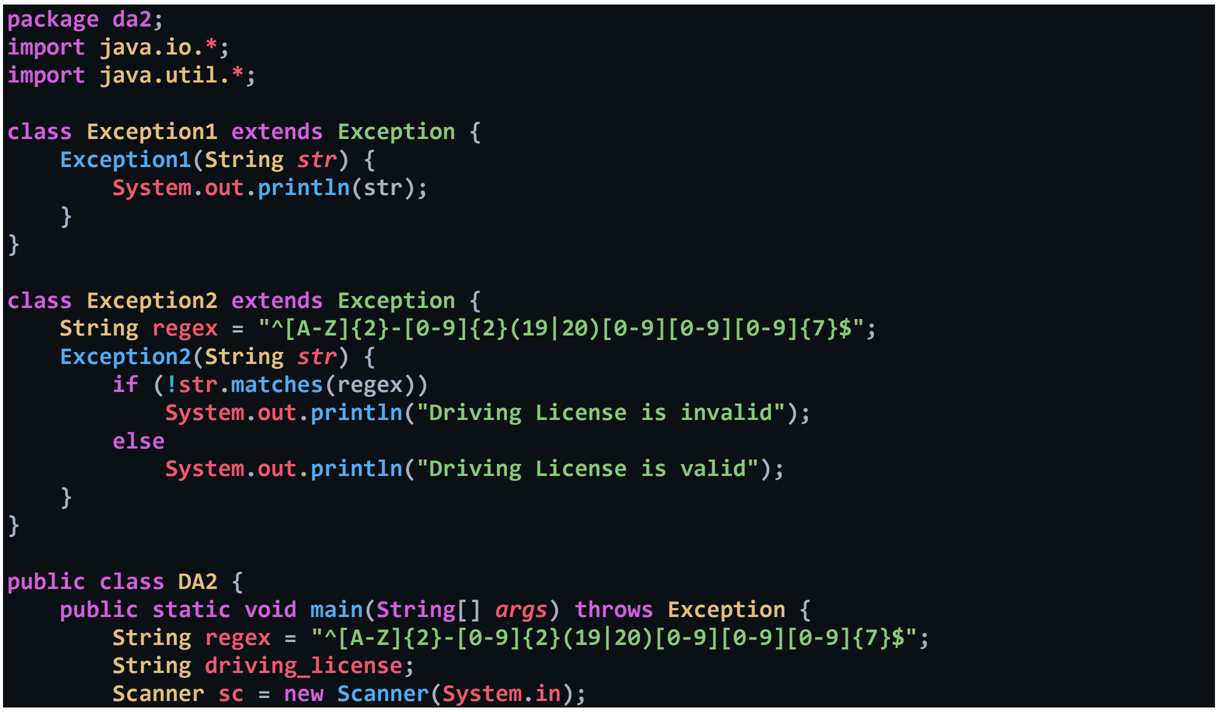
(Deemed to be University under section 3 of UGC Act, 1956)

Programme	:	**BTech. CSE Core**	Semester	:	**Win 2021-22**
Course	:	**Java Programming**	Code	:	**CSE1007**
Faculty	:	**Dr. Pradeep K**	Slot	:	**D2+TD2**
Name	:	**Hariket Sukesh Kumar Sheth**	Register No.	:	**20BCE1975**

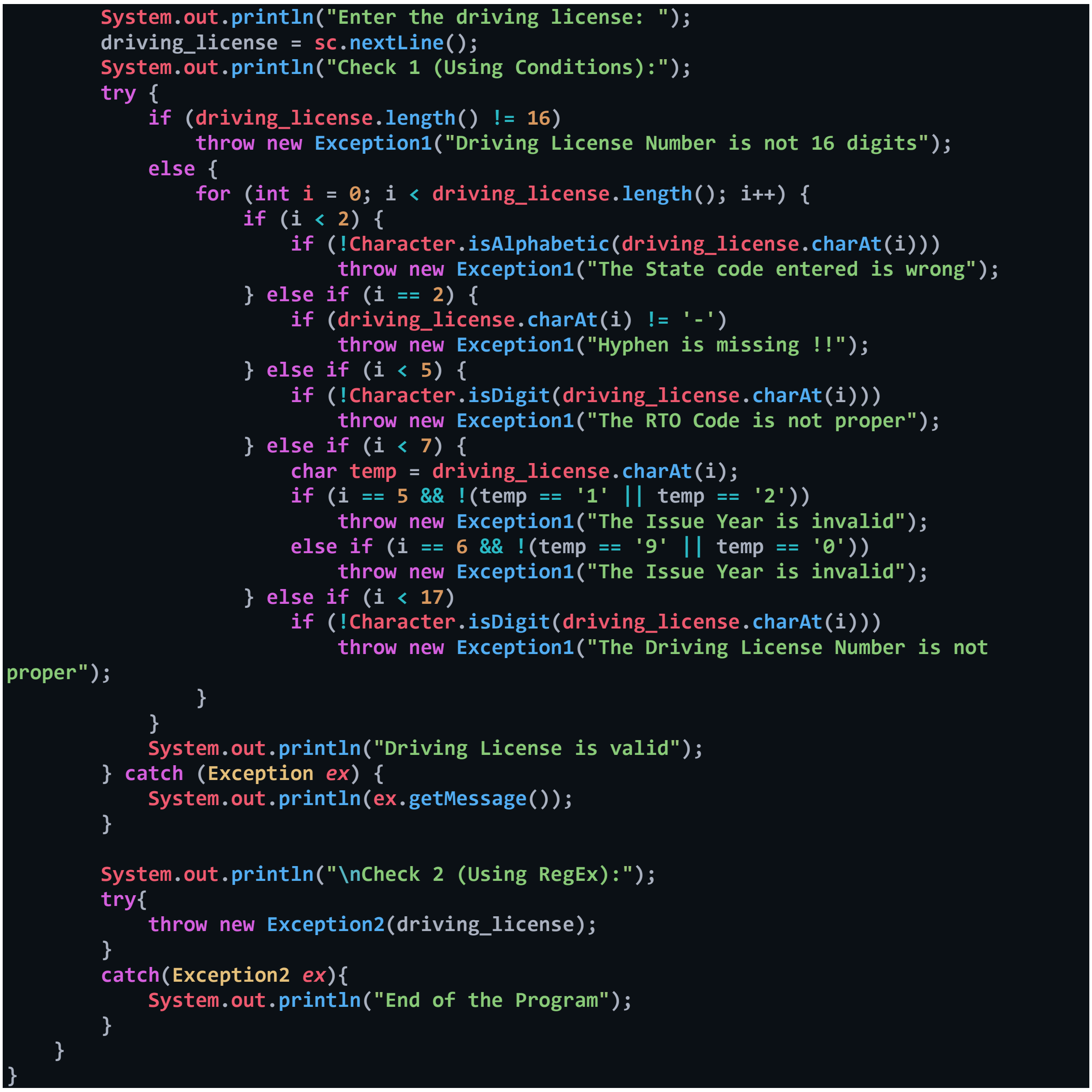
1. Write a Java Program to validate the driving License number of a citizen. The appropriate errors should be raised that are handled using Exception Handling in Java.

   The Driving License should follow the below mentioned guidelines and conditions, otherwise appropriate Exception should be raised.

   i. It should be 16 characters long (including hyphen (-)).
   ii. The driving license number can be entered in any of the following formats: CH-0119980045680
   iii. The first two characters should be upper case alphabets that represent the state code.
   iv. The next two characters should be digits that represent the RTO code.
   v. The next four characters should be digits that represent the license issued year.
   vi. The next seven characters should be any digits from 0-9.

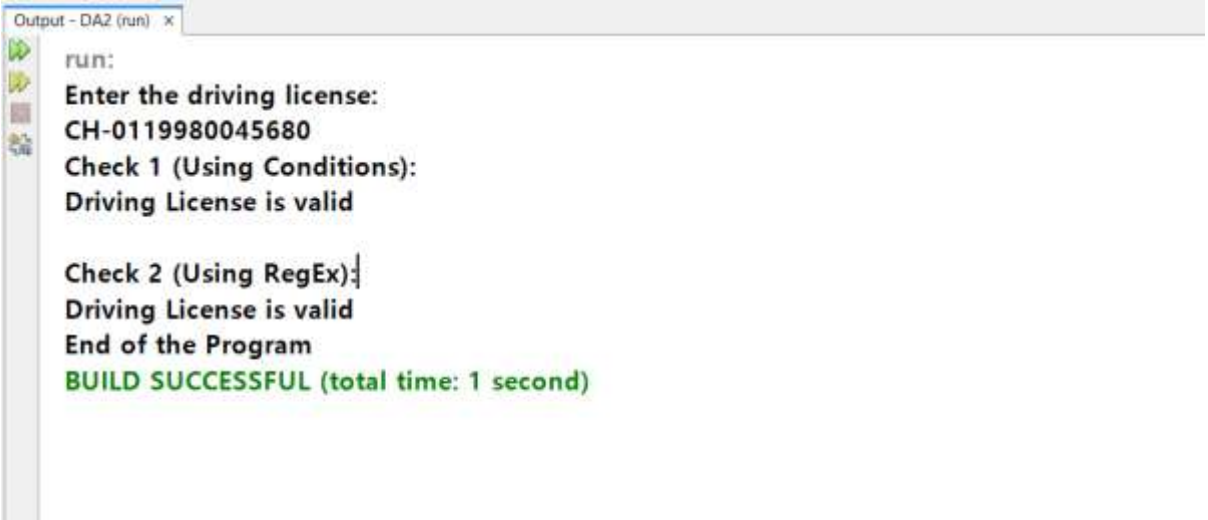


```java
package da2;
import java.io.*;
import java.util.*;

class Exception1 extends Exception {
    Exception1(String str) {
        System.out.println(str);
    }
}

class Exception2 extends Exception {
    String regex = "^[A-Z]{2}-[0-9]{2}(19|20)[0-9][0-9][0-9]{7}$";
    Exception2(String str) {
        if (!str.matches(regex))
            System.out.println("Driving License is invalid");
        else
            System.out.println("Driving License is valid");
    }
}

public class DA2 {
    public static void main(String[] args) throws Exception {
        String regex = "^[A-Z]{2}-[0-9]{2}(19|20)[0-9][0-9][0-9]{7}$";
        String driving_license;
        Scanner sc = new Scanner(System.in);
```

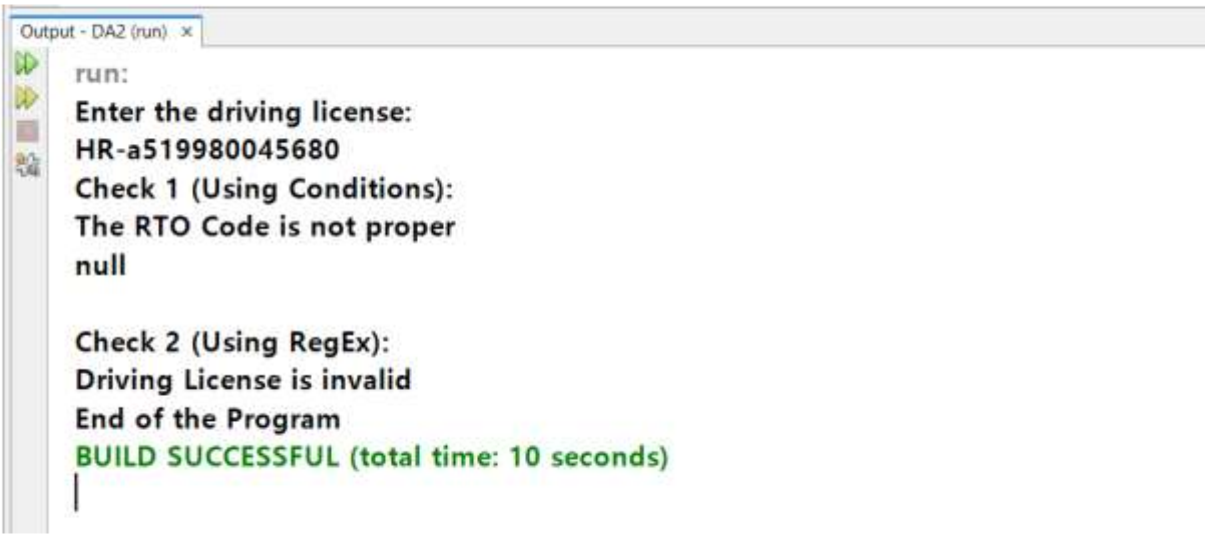


```
        System.out.println("Enter the driving license: ");
        driving_license = sc.nextLine();
        System.out.println("Check 1 (Using Conditions):");
        try {
            if (driving_license.length() != 16)
                throw new Exception1("Driving License Number is not 16 digits");
            else {
                for (int i = 0; i < driving_license.length(); i++) {
                    if (i < 2) {
                        if (!Character.isAlphabetic(driving_license.charAt(i)))
                            throw new Exception1("The State code entered is wrong");
                    } else if (i == 2) {
                        if (driving_license.charAt(i) != '-')
                            throw new Exception1("Hyphen is missing !!");
                    } else if (i < 5) {
                        if (!Character.isDigit(driving_license.charAt(i)))
                            throw new Exception1("The RTO Code is not proper");
                    } else if (i < 7) {
                        char temp = driving_license.charAt(i);
                        if (i == 5 && !(temp == '1' || temp == '2'))
                            throw new Exception1("The Issue Year is invalid");
                        else if (i == 6 && !(temp == '9' || temp == '0'))
                            throw new Exception1("The Issue Year is invalid");
                    } else if (i < 17)
                        if (!Character.isDigit(driving_license.charAt(i)))
                            throw new Exception1("The Driving License Number is not
proper");
                }
            }
            System.out.println("Driving License is valid");
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }

        System.out.println("\nCheck 2 (Using RegEx):");
        try{
            throw new Exception2(driving_license);
        }
        catch(Exception2 ex){
            System.out.println("End of the Program");
        }
    }
}
```

OUTPUT:

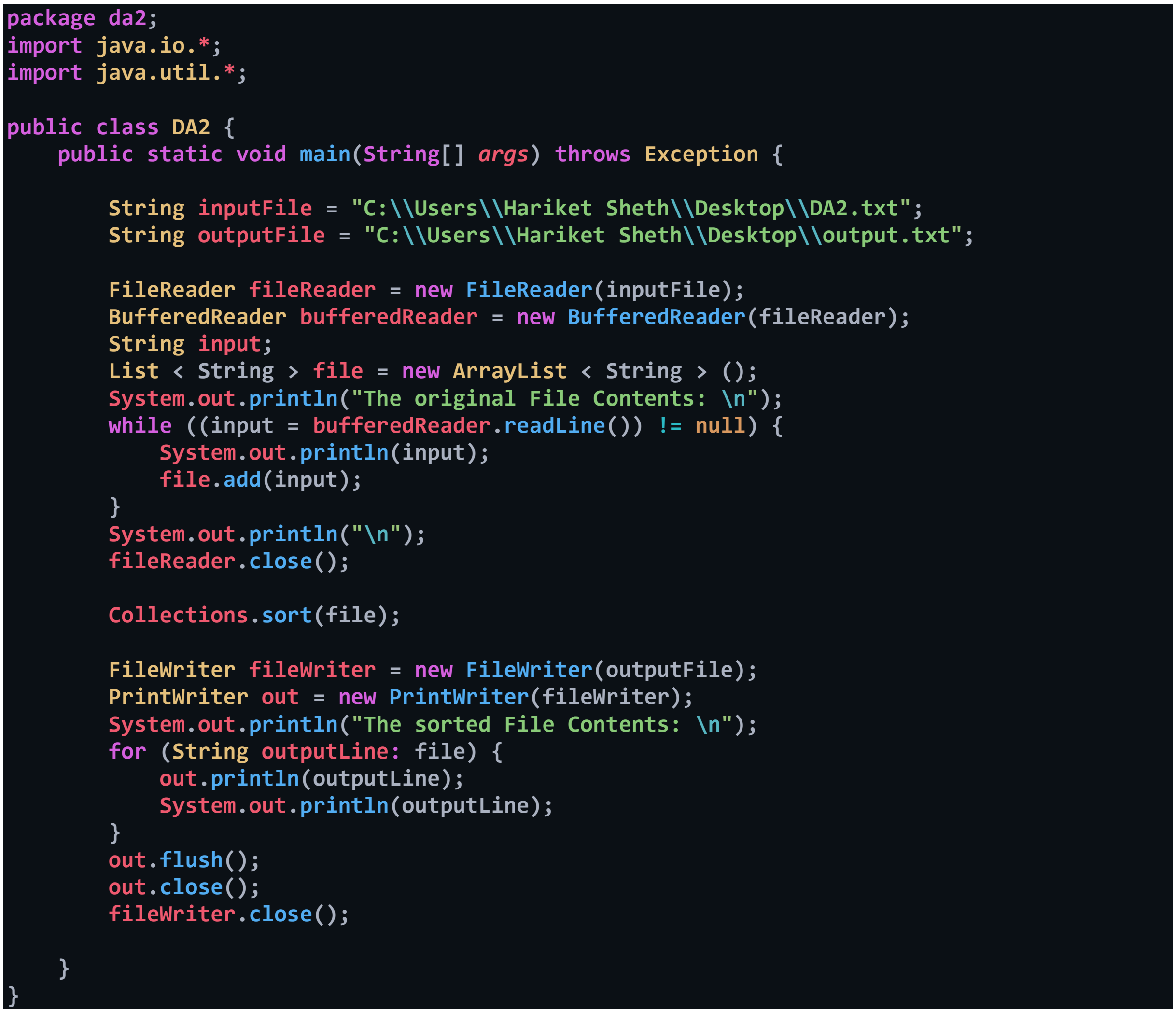


```
run:
Enter the driving license:
CH-0119980045680
Check 1 (Using Conditions):
Driving License is valid

Check 2 (Using RegEx):
Driving License is valid
End of the Program
BUILD SUCCESSFUL (total time: 1 second)
```



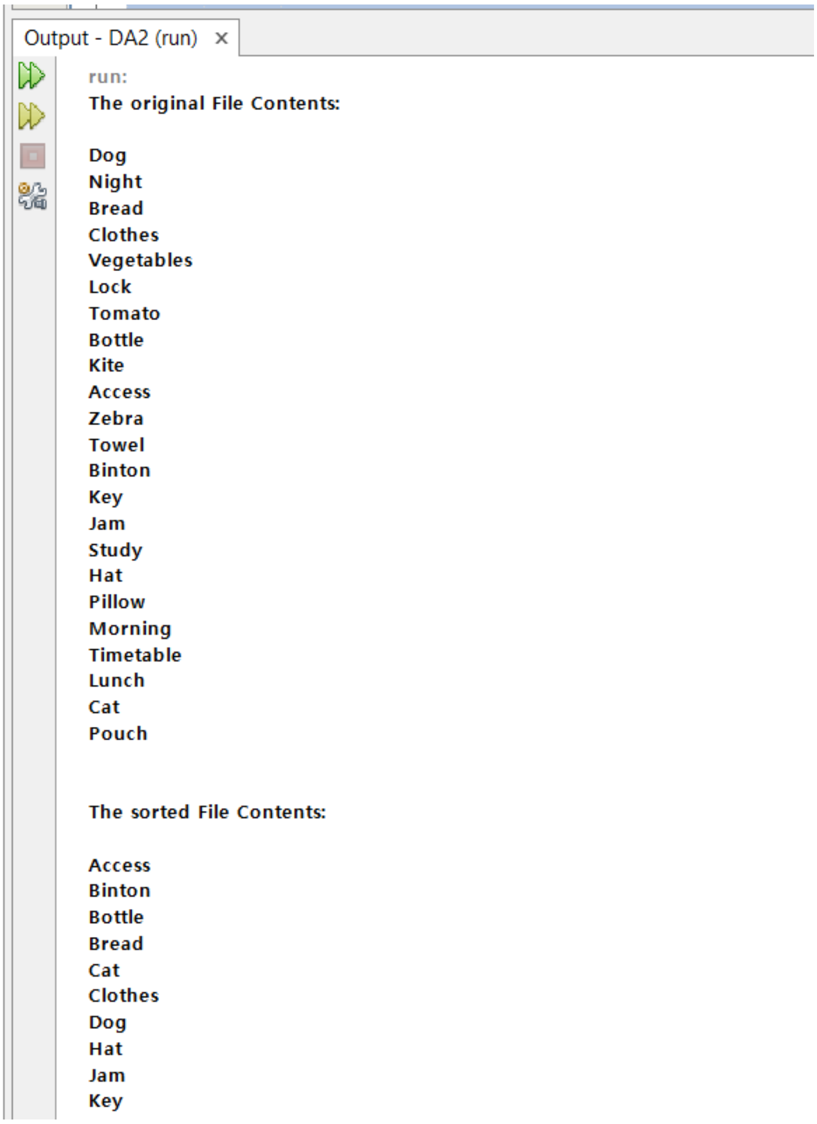
```
run:
Enter the driving license:
HR-a519980045680
Check 1 (Using Conditions):
The RTO Code is not proper
null

Check 2 (Using RegEx):
Driving License is invalid
End of the Program
BUILD SUCCESSFUL (total time: 10 seconds)
```

2. Write a Java Program to perform certain File processing Operations.
   Create A File – Read the Data From it – Print the data in Ascending Order

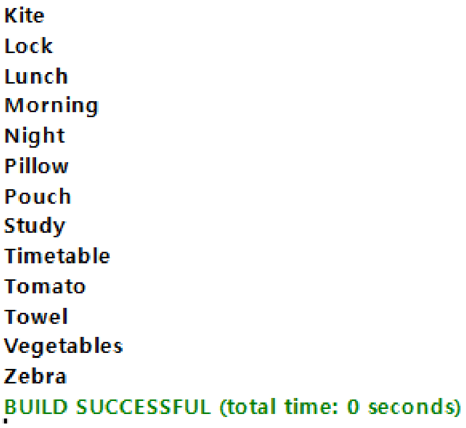


```java
package da2;
import java.io.*;
import java.util.*;

public class DA2 {
    public static void main(String[] args) throws Exception {

        String inputFile = "C:\\Users\\Hariket Sheth\\Desktop\\DA2.txt";
        String outputFile = "C:\\Users\\Hariket Sheth\\Desktop\\output.txt";

        FileReader fileReader = new FileReader(inputFile);
        BufferedReader bufferedReader = new BufferedReader(fileReader);
        String input;
        List < String > file = new ArrayList < String > ();
        System.out.println("The original File Contents: \n");
        while ((input = bufferedReader.readLine()) != null) {
            System.out.println(input);
            file.add(input);
        }
        System.out.println("\n");
        fileReader.close();

        Collections.sort(file);

        FileWriter fileWriter = new FileWriter(outputFile);
        PrintWriter out = new PrintWriter(fileWriter);
        System.out.println("The sorted File Contents: \n");
        for (String outputLine: file) {
            out.println(outputLine);
            System.out.println(outputLine);
        }
        out.flush();
        out.close();
        fileWriter.close();

    }
}
```

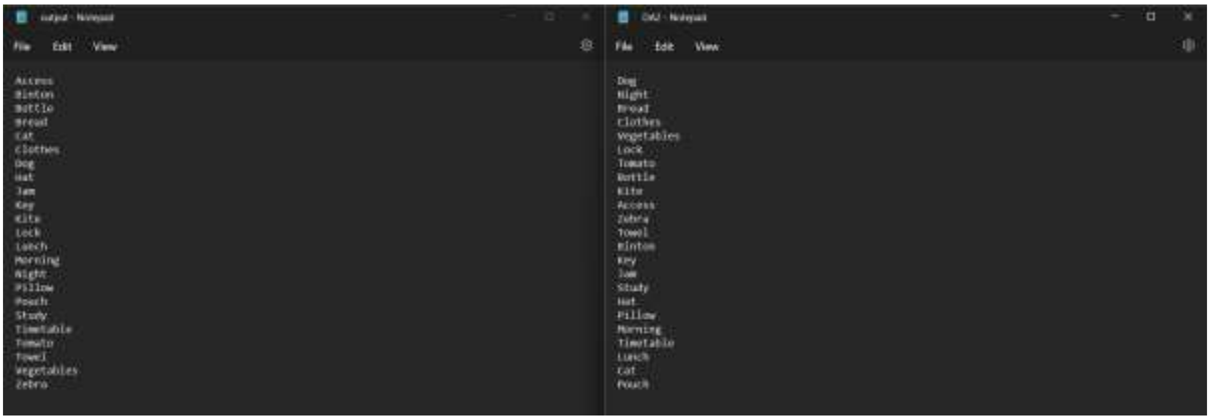
OUTPUT:



```
run:
The original File Contents:

Dog
Night
Bread
Clothes
Vegetables
Lock
Tomato
Bottle
Kite
Access
Zebra
Towel
Binton
Key
Jam
Study
Hat
Pillow
Morning
Timetable
Lunch
Cat
Pouch


The sorted File Contents:

Access
Binton
Bottle
Bread
Cat
Clothes
Dog
Hat
Jam
Key
```

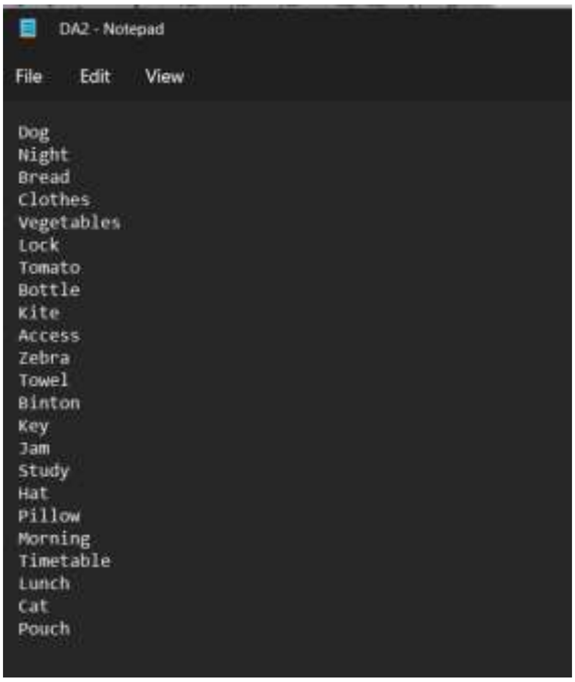


```
Kite
Lock
Lunch
Morning
Night
Pillow
Pouch
Study
Timetable
Tomato
Towel
Vegetables
Zebra
BUILD SUCCESSFUL (total time: 0 seconds)
```

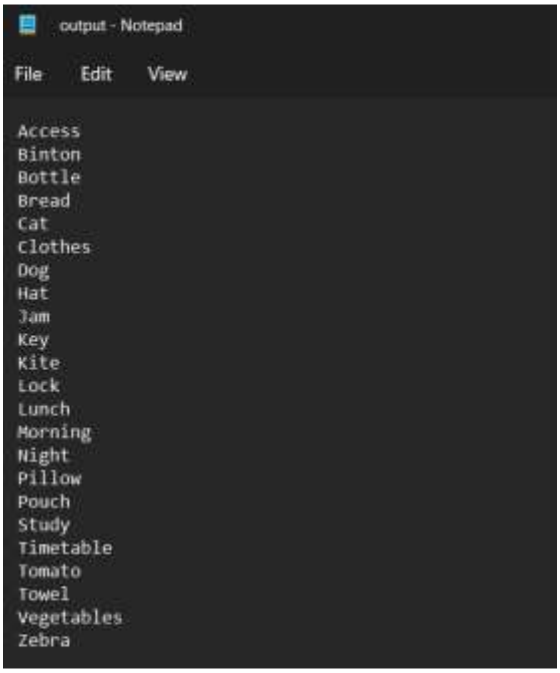


```
Access
Binton
Bottle
Bread
Cat
Clothes
Dog
Hat
Jam
Key
Kite
Lock
Lunch
Morning
Night
Pillow
Pouch
Study
Timetable
Tomato
Towel
Vegetables
Zebra
```

```
Dog
Night
Bread
Clothes
Vegetables
Lock
Tomato
Bottle
Kite
Access
Zebra
Towel
Binton
Key
Jam
Study
Hat
Pillow
Morning
Timetable
Lunch
Cat
Pouch
```



```
Dog
Night
Bread
Clothes
Vegetables
Lock
Tomato
Bottle
Kite
Access
Zebra
Towel
Binton
Key
Jam
Study
Hat
Pillow
Morning
Timetable
Lunch
Cat
Pouch
```



```
Access
Binton
Bottle
Bread
Cat
Clothes
Dog
Hat
Jam
Key
Kite
Lock
Lunch
Morning
Night
Pillow
Pouch
Study
Timetable
Tomato
Towel
Vegetables
Zebra
```