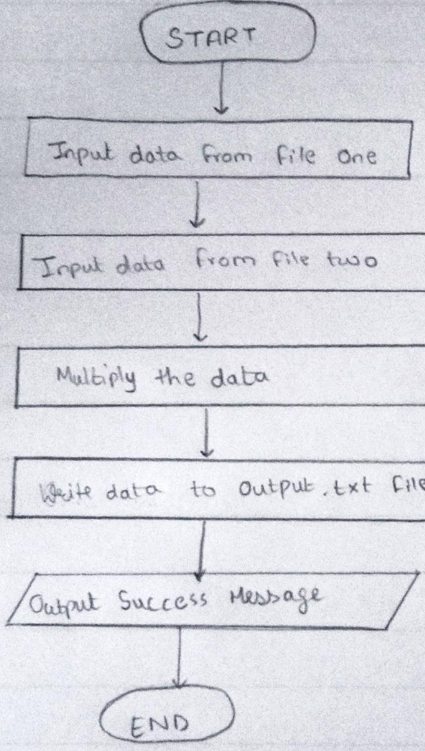
Practical No. 18

Aim : Create , debug and run java program based on read and write characters from a file using input/output stream.

Flowchart :



Practical No. 18

Aim: Create , debug and run java programs based on read and write characters from a file using input / output stream.

Theory :

The File class from the java.io package, allows us to work with files.

To use Files, we need to create object of the File class, and specify the filename or directory name

Example,

```
    import java.io.File;    OR    import java.io.*;
  File  file  =  new  File("Filename.txt");
```

The File class contains the following methods:

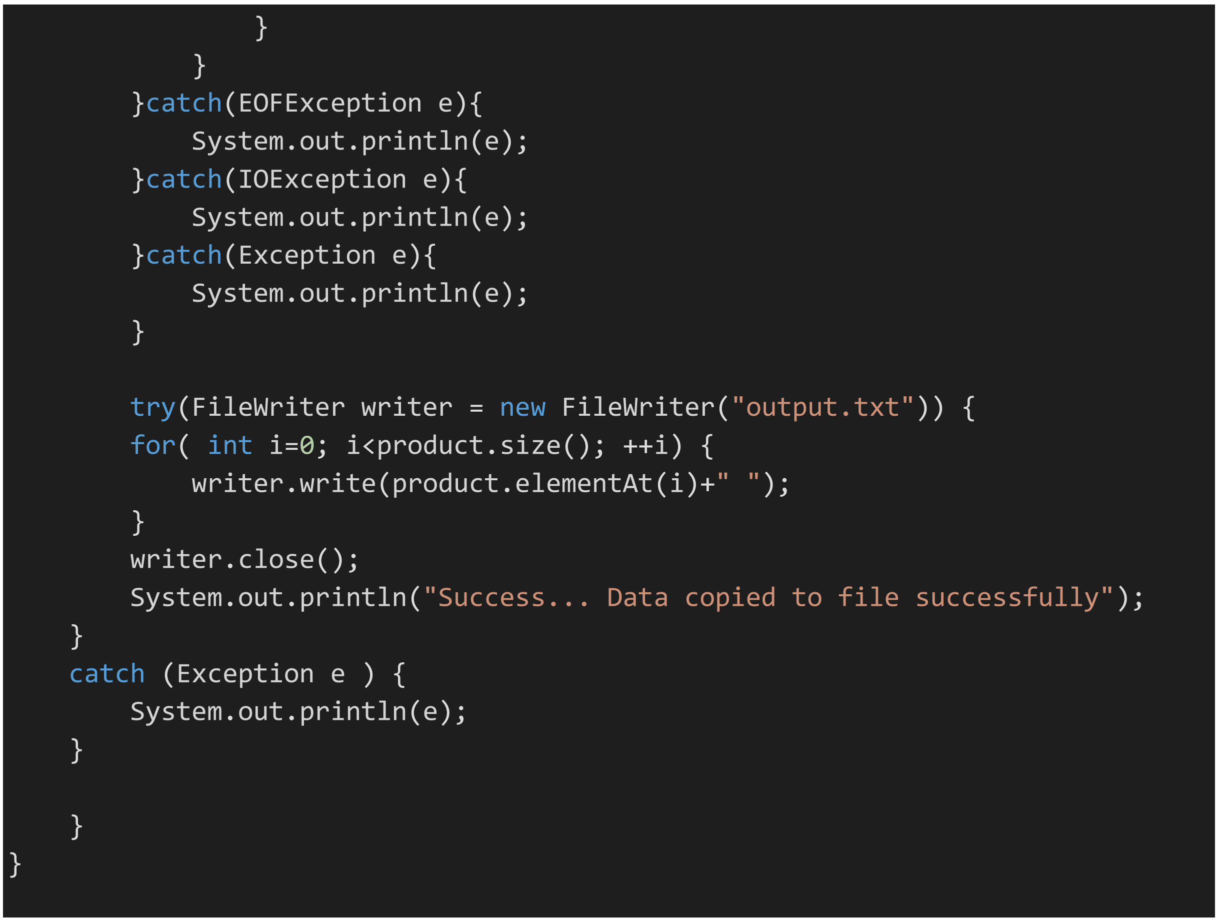
Method	Return Value	Description
canRead()	Boolean	Tests whether the file is readable or not
canWrite()	Boolean	Tests whether the file is writable or not.
createNewFile()	Boolean	Creates an empty file.
delete()	Boolean	Deletes a file
exists()	Boolean	Tests whether file exists
getName()	String	Returns the name of the file.
getAbsolutePath()	String	Returns the absolute pathname of the file.
length()	Long	Returns the size of files in bytes
list()	String[]	Returns an array of the files in the directory
mkdir()	Boolean	Create a directory.

Conclusion :

Hence, by performing this practical I get to know about the concepts of Files and performing I/O operations of them. I also created, debugged and executed Java programs based on reading and writing characters from a file using input/output stream.

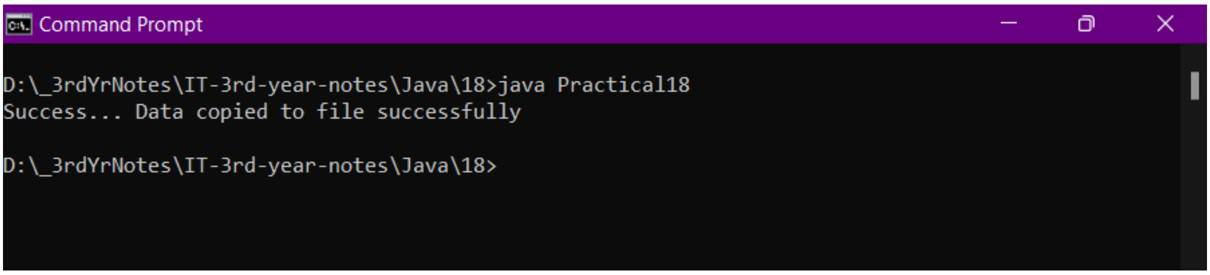
Code:

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

class Practical18{
    public static void main(String[] args){
        File input = new File("input.txt");
        File input2 = new File("input2.txt");
        File output = new File("output.txt");

            Vector<Integer> product = new Vector<Integer>();

        try{

            FileReader reader = new FileReader(input);
            Vector<Integer> v1 = new Vector<Integer>();
            Vector<Integer> v2 = new Vector<Integer>();
            int ch;
            int temp;
            while((temp = reader.read()) != -1){
                v1.add(temp);
            }
            reader.close();
            reader = new FileReader(input2);
            while((temp = reader.read()) != -1){
                v2.add(temp);
            }
            reader.close();

            int n = v1.size() > v2.size() ? v2.size() : v1.size();

            for(int i = 0; i < n; i++){
                product.add(v1.elementAt(i) * v2.elementAt(i));
            }

            if(v1.size() != v2.size()){
                if(v1.size() > v2.size()){
                    int l = v1.size() - n;
                    for(int i = n; i < l; i++){
                        product.add(v1.elementAt(i));
                    }
                }else{
                    int l = v2.size()-n;
                    for(int i = n; i < l; i++){
                        product.add(v2.elementAt(i));
                    }
```



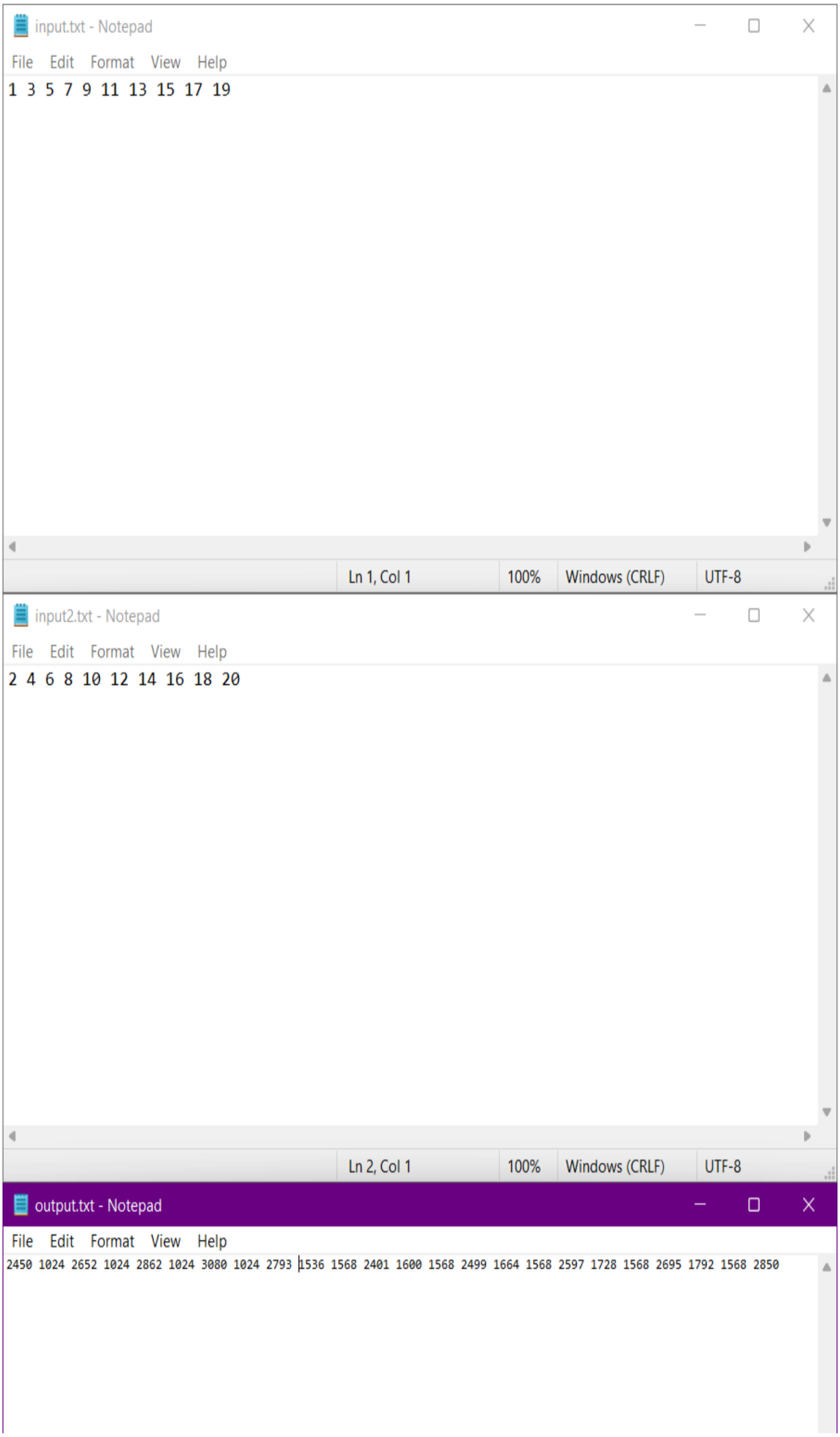
```
                }
            }
        }catch(EOFException e){
            System.out.println(e);
        }catch(IOException e){
            System.out.println(e);
        }catch(Exception e){
            System.out.println(e);
        }

        try(FileWriter writer = new FileWriter("output.txt")) {
        for( int i=0; i<product.size(); ++i) {
            writer.write(product.elementAt(i)+" ");
        }
        writer.close();
        System.out.println("Success... Data copied to file successfully");
    }
    catch (Exception e ) {
        System.out.println(e);
    }

    }
}
```

Output:



```
D:\_3rdYrNotes\IT-3rd-year-notes\Java\18>java Practical18
Success... Data copied to file successfully

D:\_3rdYrNotes\IT-3rd-year-notes\Java\18>
```

Output on terminal



```
1 3 5 7 9 11 13 15 17 19
```

```
2 4 6 8 10 12 14 16 18 20
```

```
2450 1024 2652 1024 2862 1024 3080 1024 2793 1536 1568 2401 1600 1568 2499 1664 1568 2597 1728 1568 2695 1792 1568 2850
```

Input files and the output file

Conclusion :

Hence, by performing this practical I got to know about the concepts of files and Performing I/O operations on them. I also created, debug and executed Java programs based on reading and writing characters from a file using input / output stream.