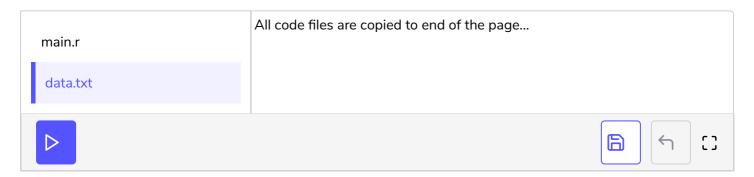
Solution Review: Handling TXT files

In this review, we provide a detailed analysis of the solution to this problem.



Solution: Input/Output from File



Explanation

This exercise is similar to exercise 13, however, now we have to take input from a file and also output on file.

The main execution of the code given above starts from **line number 20**. We first open the file located at the given <code>path</code>. We then read all the lines present in the file and store in <code>myVector</code>. Next, we pass this vector to the function <code>evenOdd()</code>. This function returns us the <code>myOutputVector</code> vector, which we write on file <code>output/outData.txt</code>.

In the next lesson, we will learn how to handle .csv files.

Code Files Content !!!

```
| main.r [1]
evenOdd <- function(myVector) # a function that returns a vector</pre>
                       # whose each element tells even or odd corresponding to input vector
{
  myOutputVector <- vector("character", 0)</pre>
  for(i in myVector)
      if(as.integer(i) \%\% 2 == 0)
        myOutputVector <- c(myOutputVector, "even")</pre>
      else
      {
        myOutputVector <- c(myOutputVector, "odd")</pre>
  }
  return(myOutputVector)
}
# Driver Code
path <- "data.txt"
fileData <- file(path, open = "r") # open the file located in path</pre>
lines <- readLines(fileData) # read lines of the file</pre>
myVector <- vector("numeric", 0) # vector to store data</pre>
for (i in 1:length(lines)) # iterate over all the lines
  myVector <- c(myVector, lines[i]) # append lines in myVector</pre>
}
result <- evenOdd(myVector) # store output in result</pre>
write(result, "output/outData.txt") # write result on file
data.txt [1]
1
2
3
```