****

Sri Lanka Institute of Information Technology

**User Guide**

Complexity Measuring Tool

Software Project Management

Group ID :- SPM\_19\_RSE\_WE\_15

Submitted by:

1. IT16178700 – D.I.K. Rajapakshe
2. IT17018760 – M.P.P. Shamil
3. IT17152938 – S.K. Liyanage
4. IT17056212 – P.M.C.P. Paththinisekara

Date of submission

**Introduction to the tool**

This tool is designed to measure software complexity of a given program as an attempt to reduce the maintenance cost of the software development. Once the code of the program uploaded to the tool it will automatically calculate the code complexity using various factors such as,

**a. Size**

**b. Type and the nesting level of control structures**

**c. Inheritance**

**d. Recursion**

Then it will calculate the Total weight of the program (TW), compute the complexity of a program statement (Cps) and calculate complexity introduced due to recursion (Cr). Tool is designed to display the complexity of a program in a table according to the line by line in the code. And also, it will show the results in chart according to the main factors to make it easier for the user.

**Technical requirements**

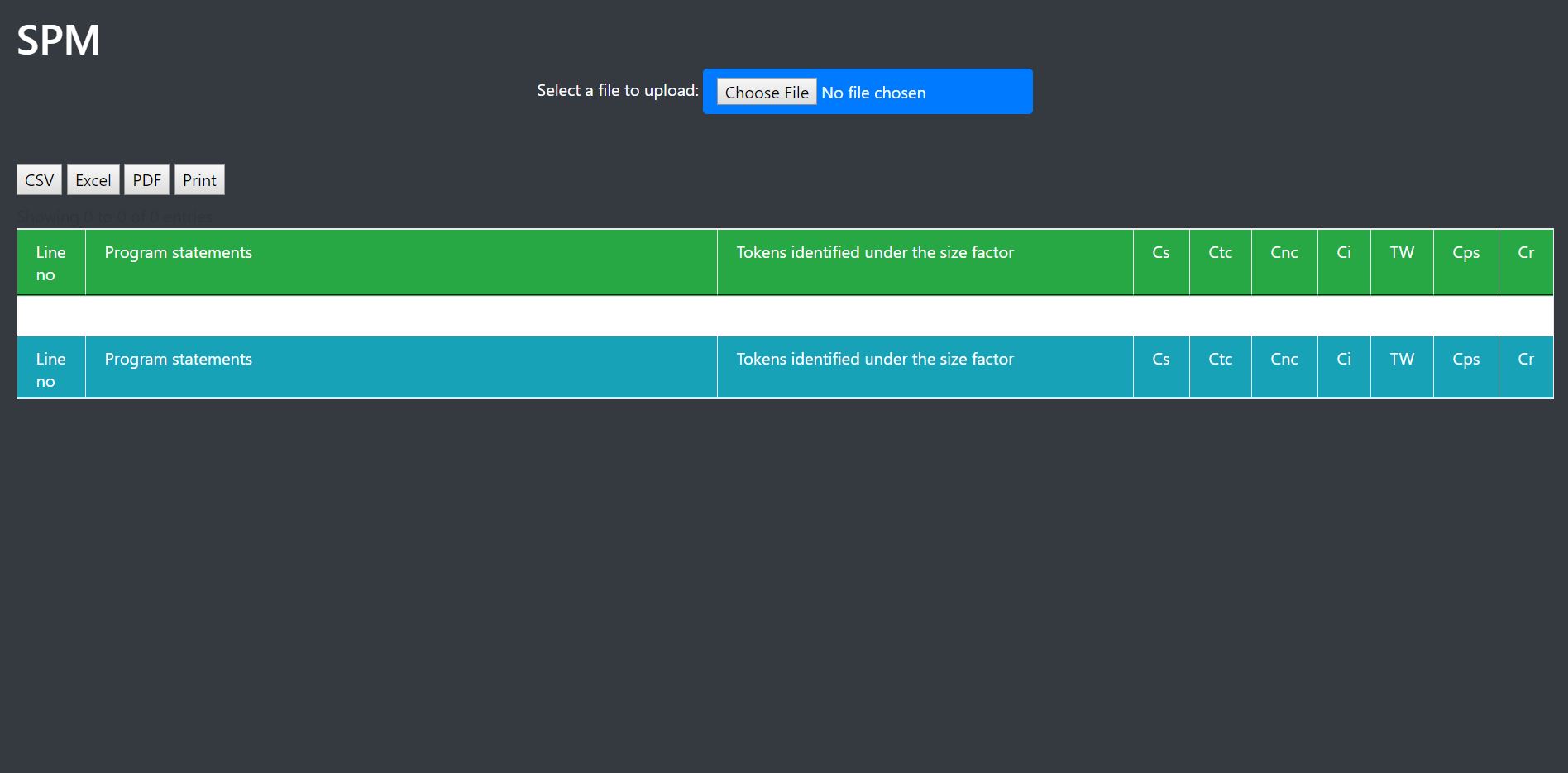
1. Java JDK 8 or above version.
2. Tomcat server 8.5 or above version.
3. This was developed using Intellij IDEA you will require Intellij IDEA to run the code.
4. Please update mavel repository in intellij IDEA project.
5. Please install all the dependencies in the intellij project **pom.xml** before run the code.

<!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->  
 <dependency>  
 <groupId>commons-fileupload</groupId>  
 <artifactId>commons-fileupload</artifactId>  
 <version>1.4</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/commons-io/commons-io -->  
 <dependency>  
 <groupId>commons-io</groupId>  
 <artifactId>commons-io</artifactId>  
 <version>2.6</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->  
 <dependency>  
 <groupId>com.google.code.gson</groupId>  
 <artifactId>gson</artifactId>  
 <version>2.8.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.12</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>RELEASE</version>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 </dependency>

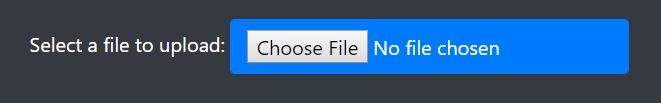
1. If you have the java 8+ runtime environment and tomcat 8.5+ version you can just execute the WAR file instead.

**Steps to use the tool**

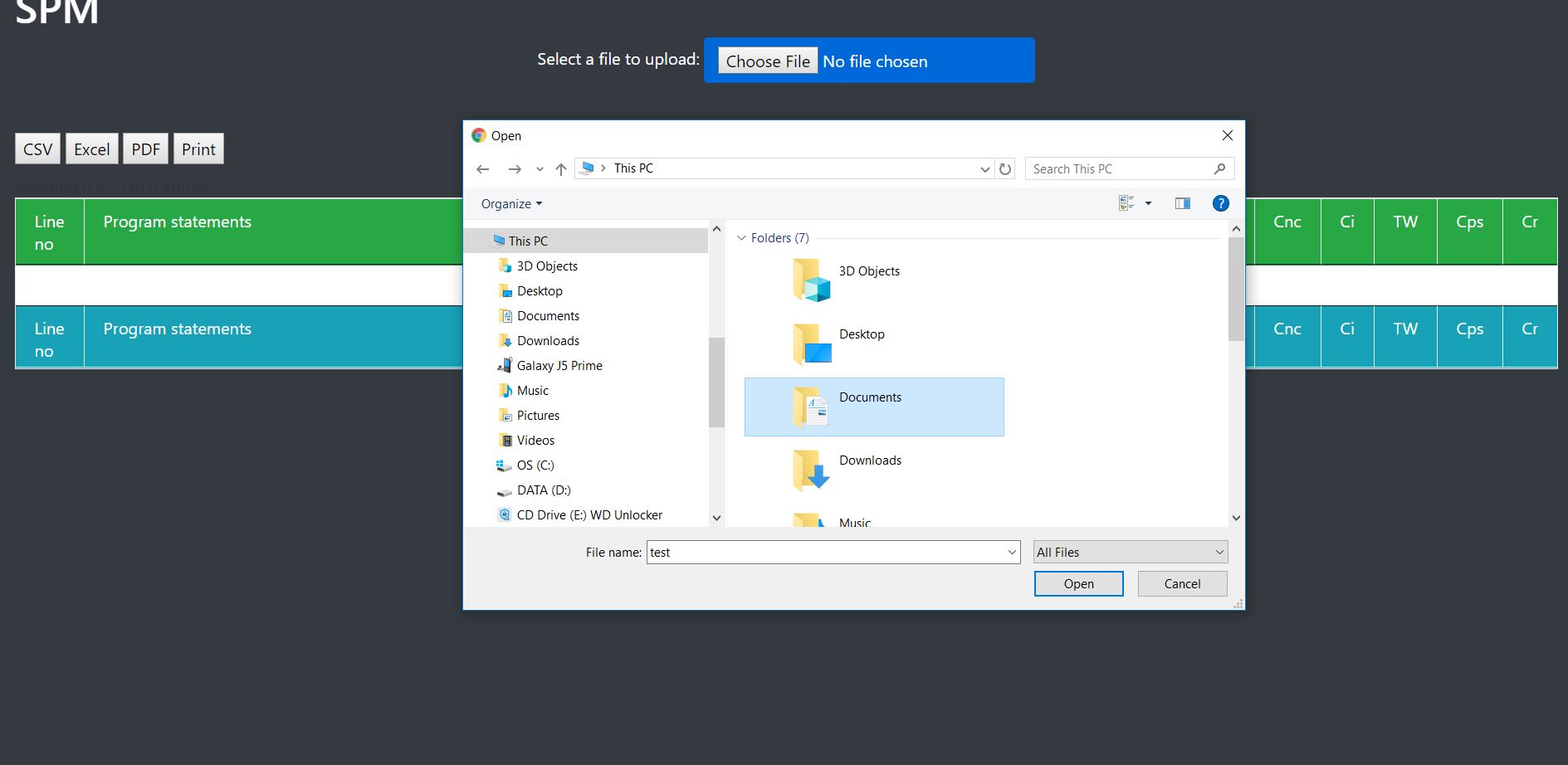
1. **Upload a File: -** First load the main screen of the tool. This page will mainly show a table and the Choose file button.

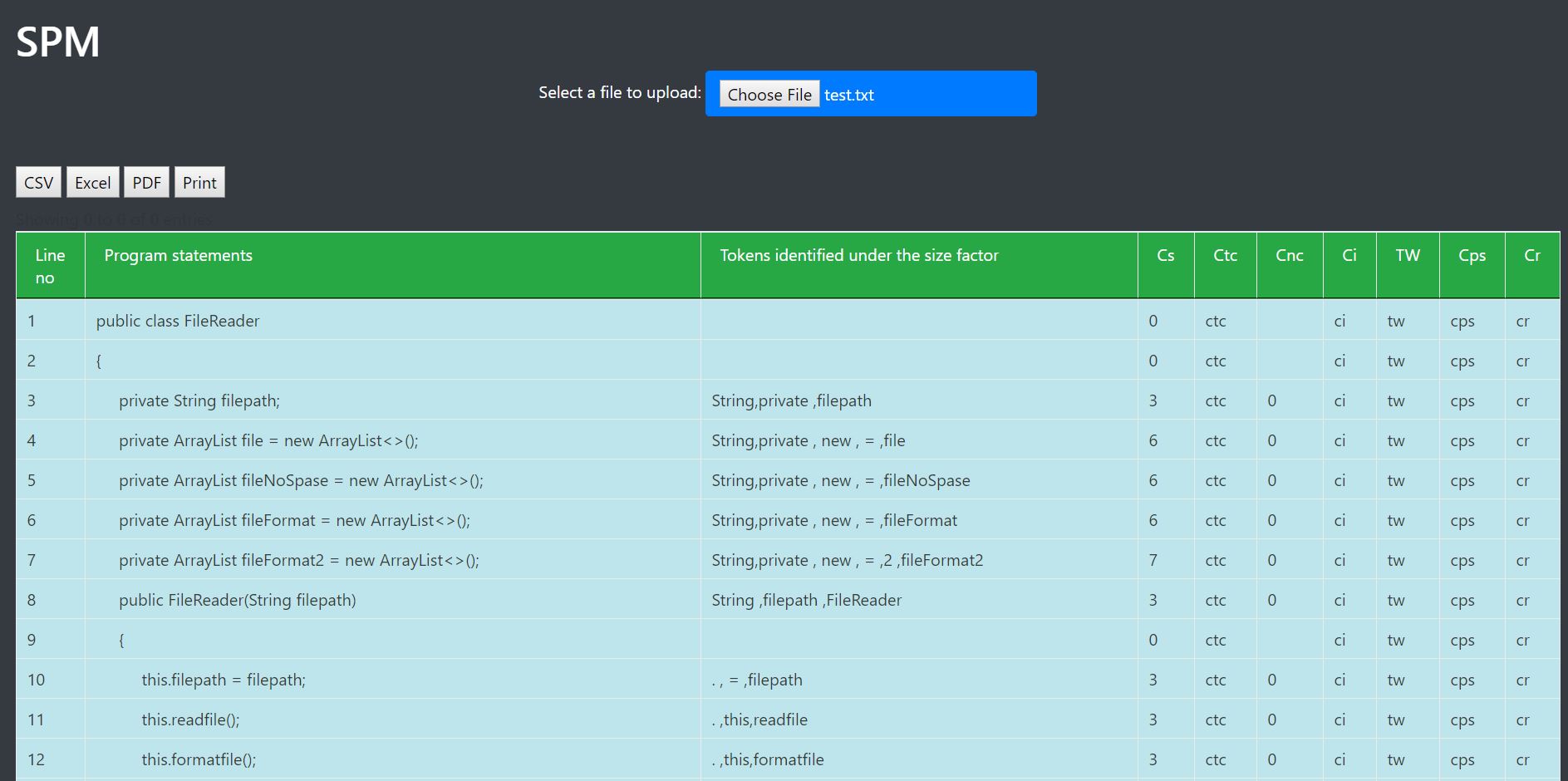
****

* Then click the “Choose File” button shows in the below picture.



* Then browse for the code file which you need to find the complexity of.

****

1. **Compute complexity: -** Then the toolwill automatically calculate and show the code complexity line by line according to the given 4 different factors.

* Using those main factors, the tool will automatically compute the complexity according to below formulas.

**Measuring total complexity of a program statement: -**

First, compute total weight (TW) of a program statement as follows:

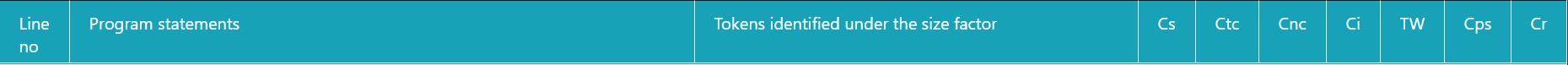
**TW = Ctc + Cnc + Ci**

Next, compute the complexity of a program statement (Cps) as follows:

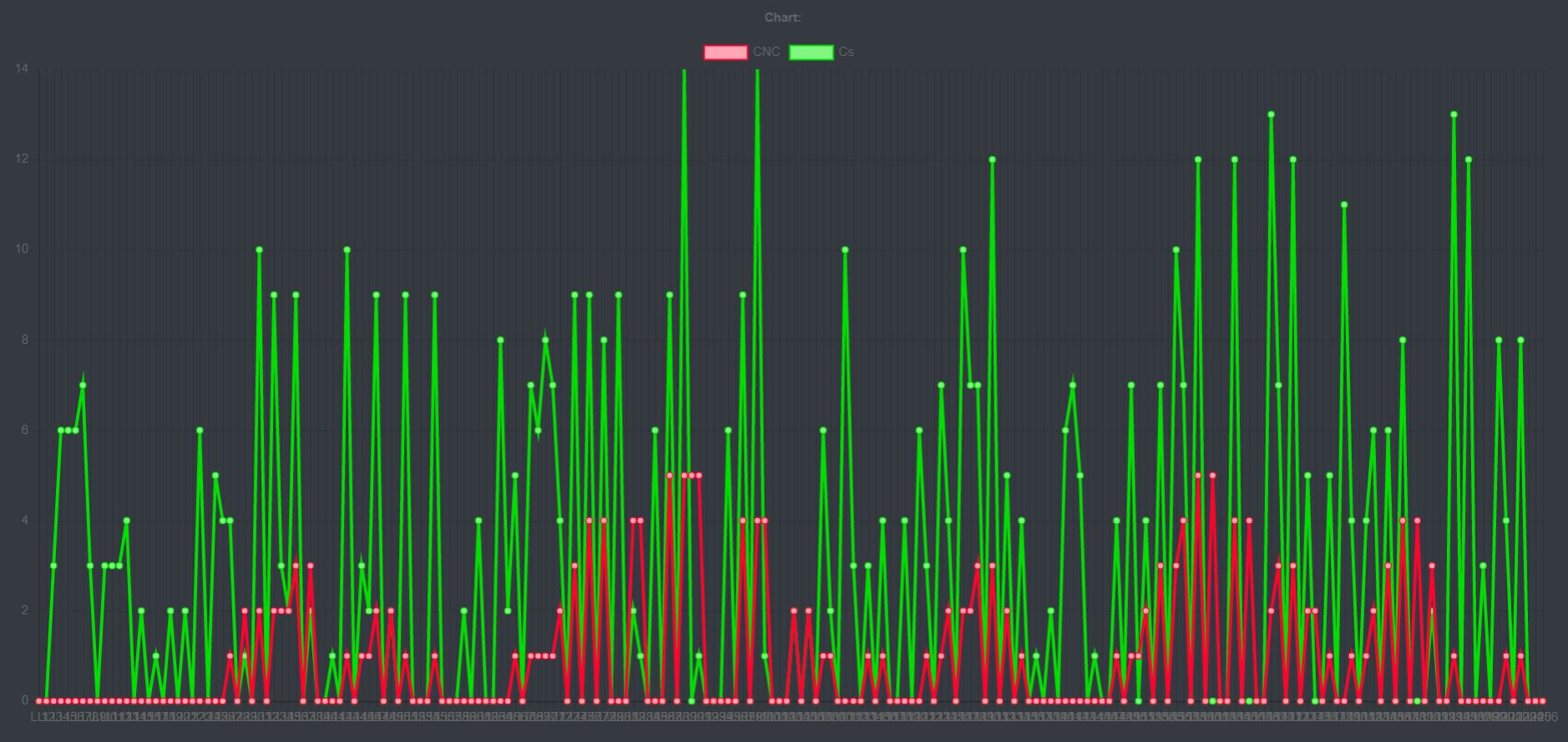
**Cps = Cs \* TW**

**Measuring the complexity introduced due to recursion (Cr): -**

Double the Cps values derived for each program statement that belongs to a recursive method.



1. **Complexity chart: -** The tool will generate a chart based on the results of the complexity table generated above.

****

1. **Print/Download the complexity table: -** You can save or print the complexity table by clicking on buttons shows below.

****