acm International Collegiate Programming Contest

IBM event sponsor

Online Judge

Online Judge

Output checking, Assembly Code checking and Code similarity

Instructor : Dr. Ali Hamzeh

Arash Pourhabibi Zarandi, Morteza Dastouri

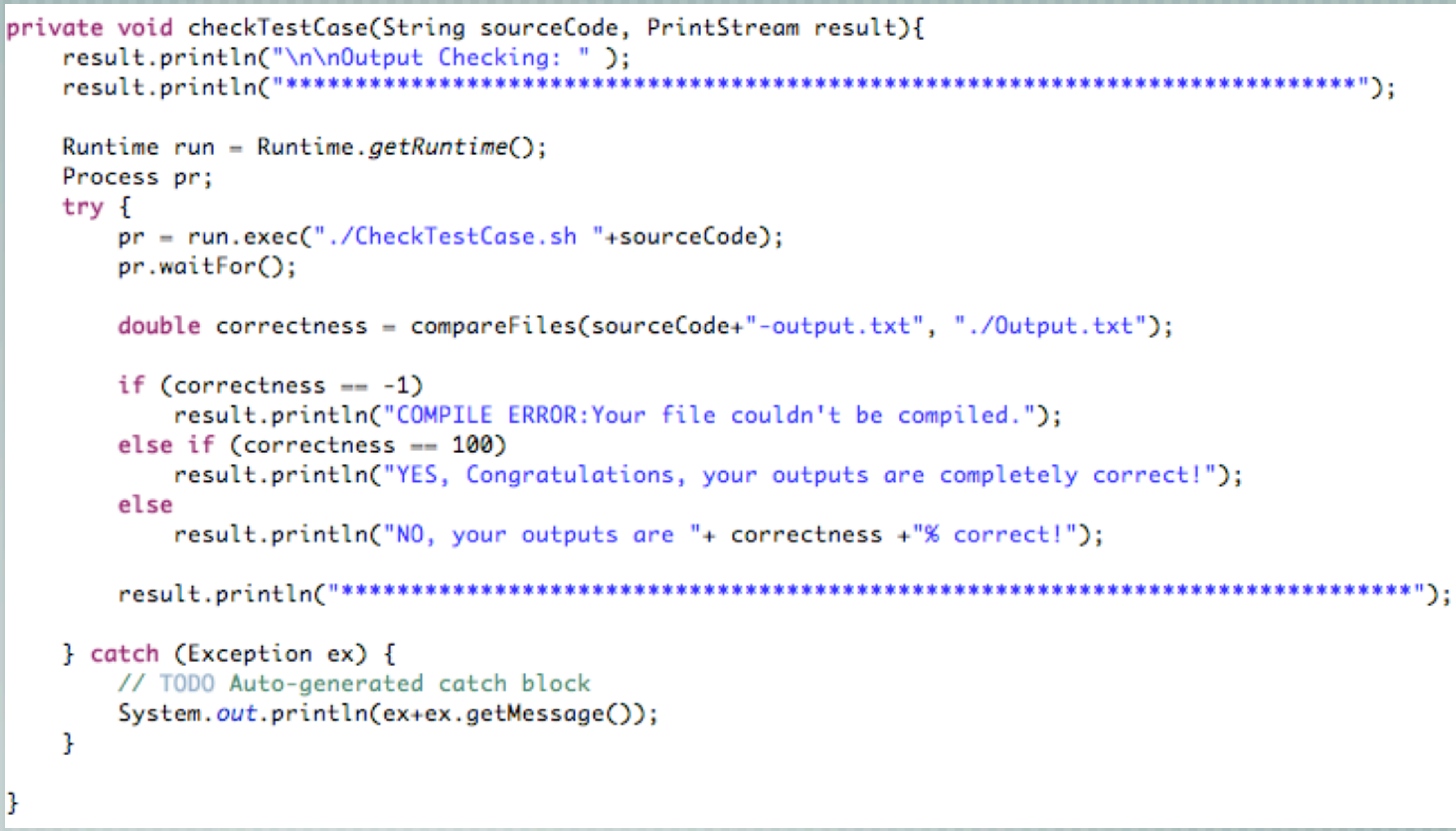
Shiraz University, School of Computer Science and Engineering, January 2011

Intro

- Checking Methods
- Socket Programming
- Multi-Client Server

Output Checking

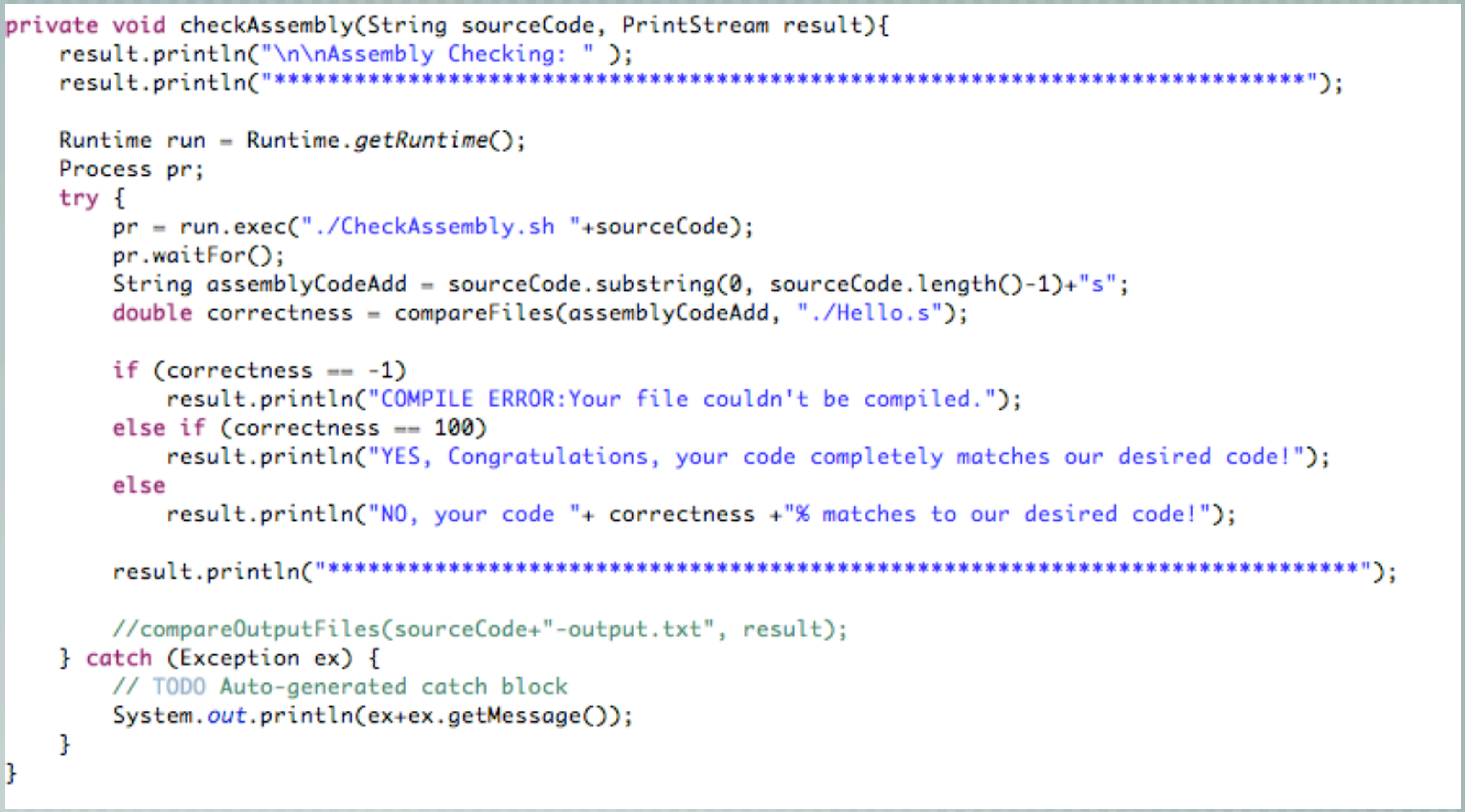
- **Using shell commands**
- **Comparing each bytes**



```
private void checkTestCase(String sourceCode, PrintStream result){
    result.println("\n\nOutput Checking: " );
    result.println("******************************************************************************");

    Runtime run = Runtime.getRuntime();
    Process pr;
    try {
        pr = run.exec("./CheckTestCase.sh "+sourceCode);
        pr.waitFor();

        double correctness = compareFiles(sourceCode+"-output.txt", "./Output.txt");

        if (correctness == -1)
            result.println("COMPILE ERROR:Your file couldn't be compiled.");
        else if (correctness == 100)
            result.println("YES, Congratulations, your outputs are completely correct!");
        else
            result.println("NO, your outputs are "+ correctness +"% correct!");

        result.println("******************************************************************************");

    } catch (Exception ex) {
        // TODO Auto-generated catch block
        System.out.println(ex+ex.getMessage());
    }

}
```

Assembly Code Checking

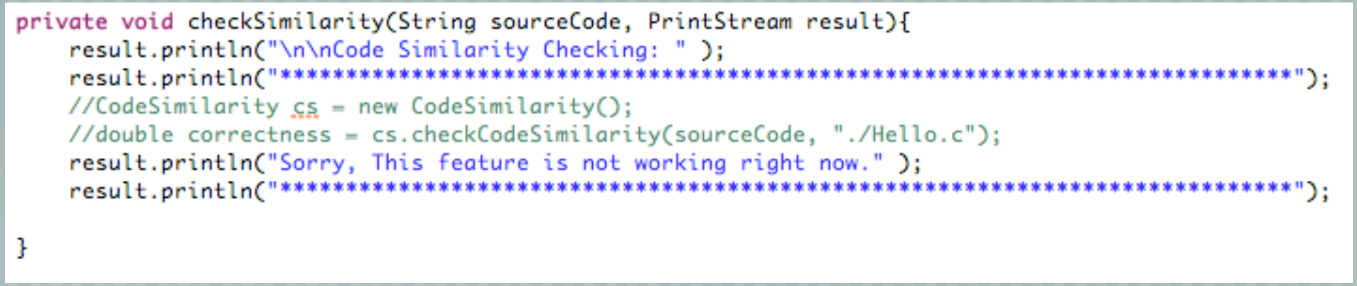
- **Using shell commands**
- **gcc -S**
- **Why Assembly?**
- **Control Flow**
- **Comparing each bytes**



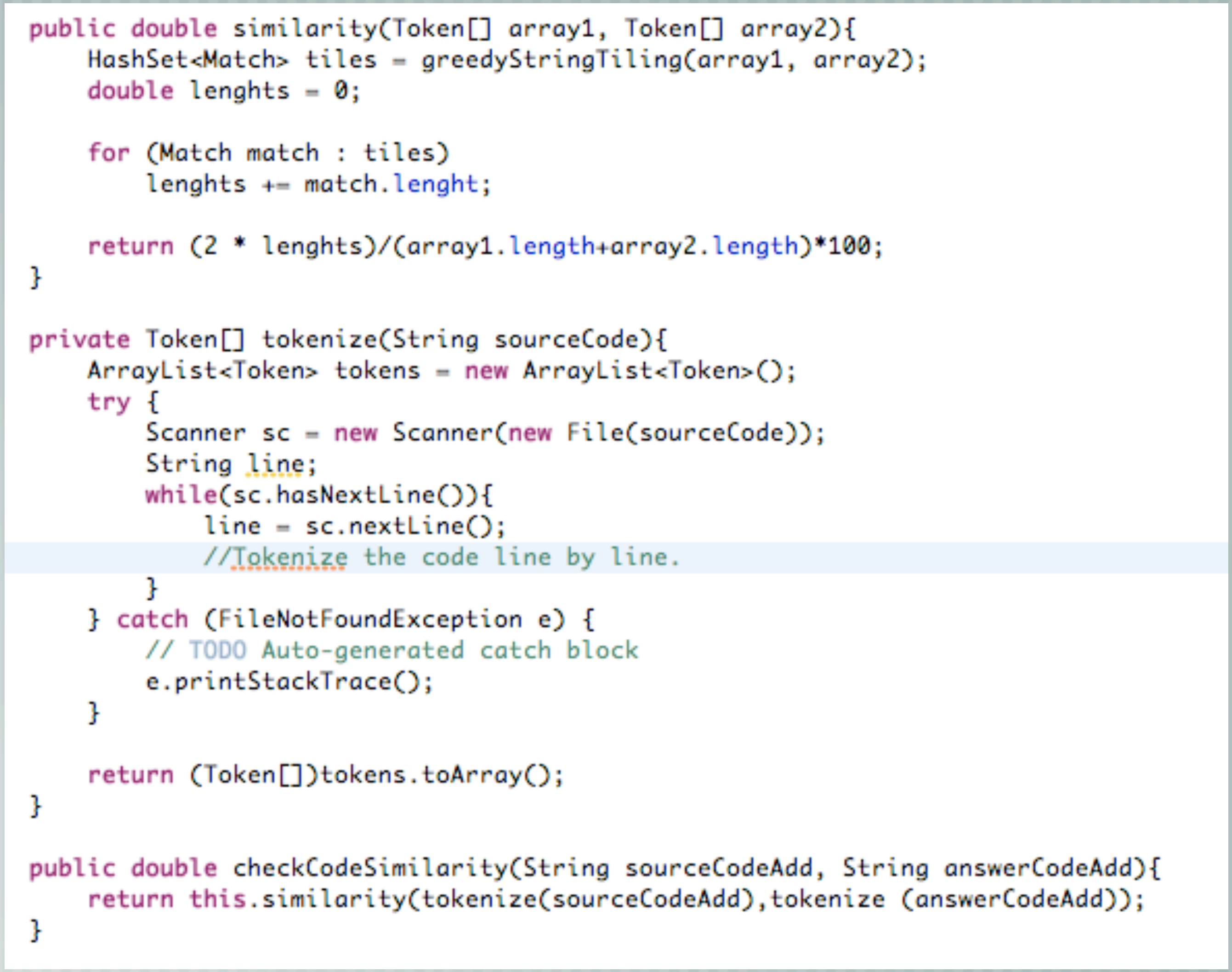
```
private void checkAssembly(String sourceCode, PrintStream result){
    result.println("\n\nAssembly Checking: " );
    result.println("*************************************************************************");

    Runtime run = Runtime.getRuntime();
    Process pr;
    try {
        pr = run.exec("./CheckAssembly.sh "+sourceCode);
        pr.waitFor();
        String assemblyCodeAdd = sourceCode.substring(0, sourceCode.length()-1)+"s";
        double correctness = compareFiles(assemblyCodeAdd, "./Hello.s");

        if (correctness == -1)
            result.println("COMPILE ERROR:Your file couldn't be compiled.");
        else if (correctness == 100)
            result.println("YES, Congratulations, your code completely matches our desired code!");
        else
            result.println("NO, your code "+ correctness +"% matches to our desired code!");

        result.println("*************************************************************************");

        //compareOutputFiles(sourceCode+"-output.txt", result);
    } catch (Exception ex) {
        // TODO Auto-generated catch block
        System.out.println(ex+ex.getMessage());
    }
}
```

Code Similarity

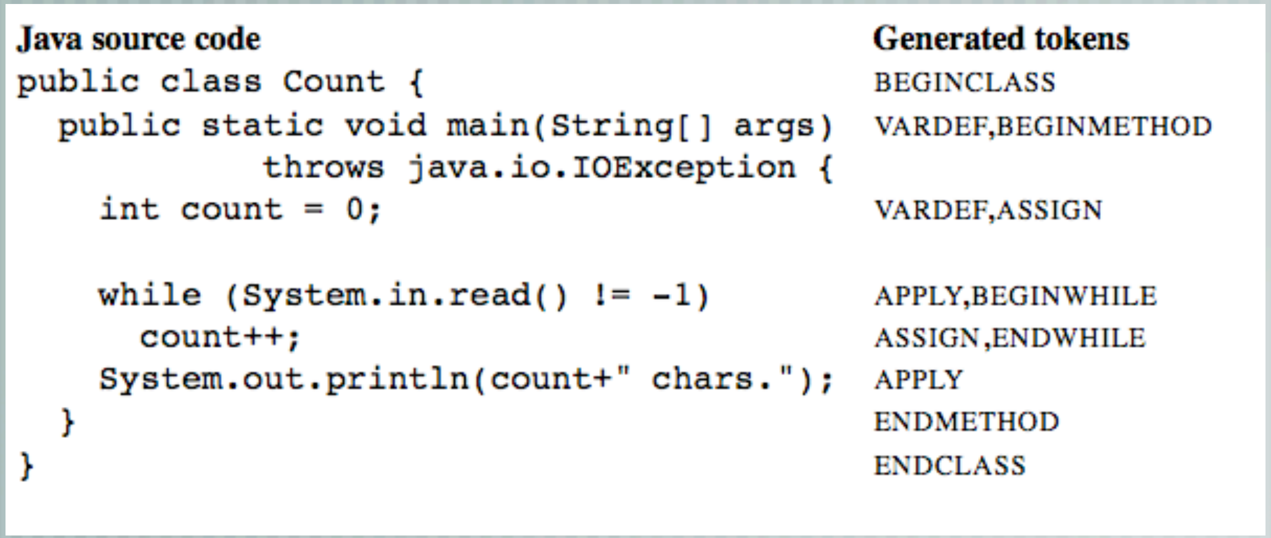
- **Plagiarism!!**
- **Code Similarity**
- **Tokenizing**



```java
private void checkSimilarity(String sourceCode, PrintStream result){
    result.println("\n\nCode Similarity Checking: " );
    result.println("*****************************************************************************");
    //CodeSimilarity cs = new CodeSimilarity();
    //double correctness = cs.checkCodeSimilarity(sourceCode, "./Hello.c");
    result.println("Sorry, This feature is not working right now." );
    result.println("*****************************************************************************");

}
```



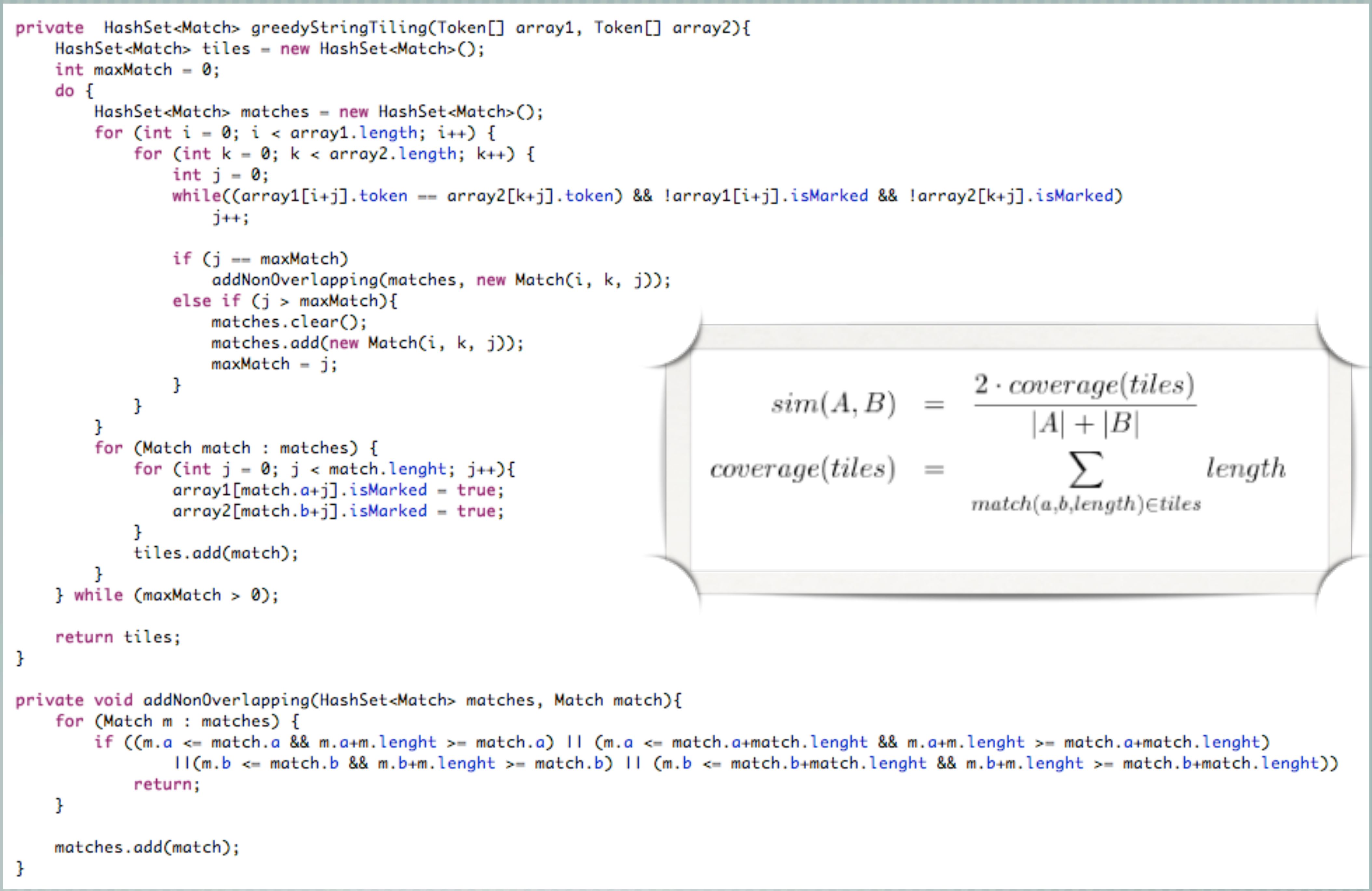
```java
public double similarity(Token[] array1, Token[] array2){
    HashSet<Match> tiles = greedyStringTiling(array1, array2);
    double lenghts = 0;

    for (Match match : tiles)
        lenghts += match.lenght;

    return (2 * lenghts)/(array1.length+array2.length)*100;
}

private Token[] tokenize(String sourceCode){
    ArrayList<Token> tokens = new ArrayList<Token>();
    try {
        Scanner sc = new Scanner(new File(sourceCode));
        String line;
        while(sc.hasNextLine()){
            line = sc.nextLine();
            //Tokenize the code line by line.
        }
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return (Token[])tokens.toArray();
}

public double checkCodeSimilarity(String sourceCodeAdd, String answerCodeAdd){
    return this.similarity(tokenize(sourceCodeAdd),tokenize (answerCodeAdd));
}
```



Java source code	Generated tokens
`public class Count {`	BEGINCLASS
`  public static void main(String[] args)`	VARDEF,BEGINMETHOD
`            throws java.io.IOException {`	
`    int count = 0;`	VARDEF,ASSIGN
`    while (System.in.read() != -1)`	APPLY,BEGINWHILE
`      count++;`	ASSIGN,ENDWHILE
`    System.out.println(count+" chars.");`	APPLY
`  }`	ENDMETHOD
`}`	ENDCLASS

Code Similarity

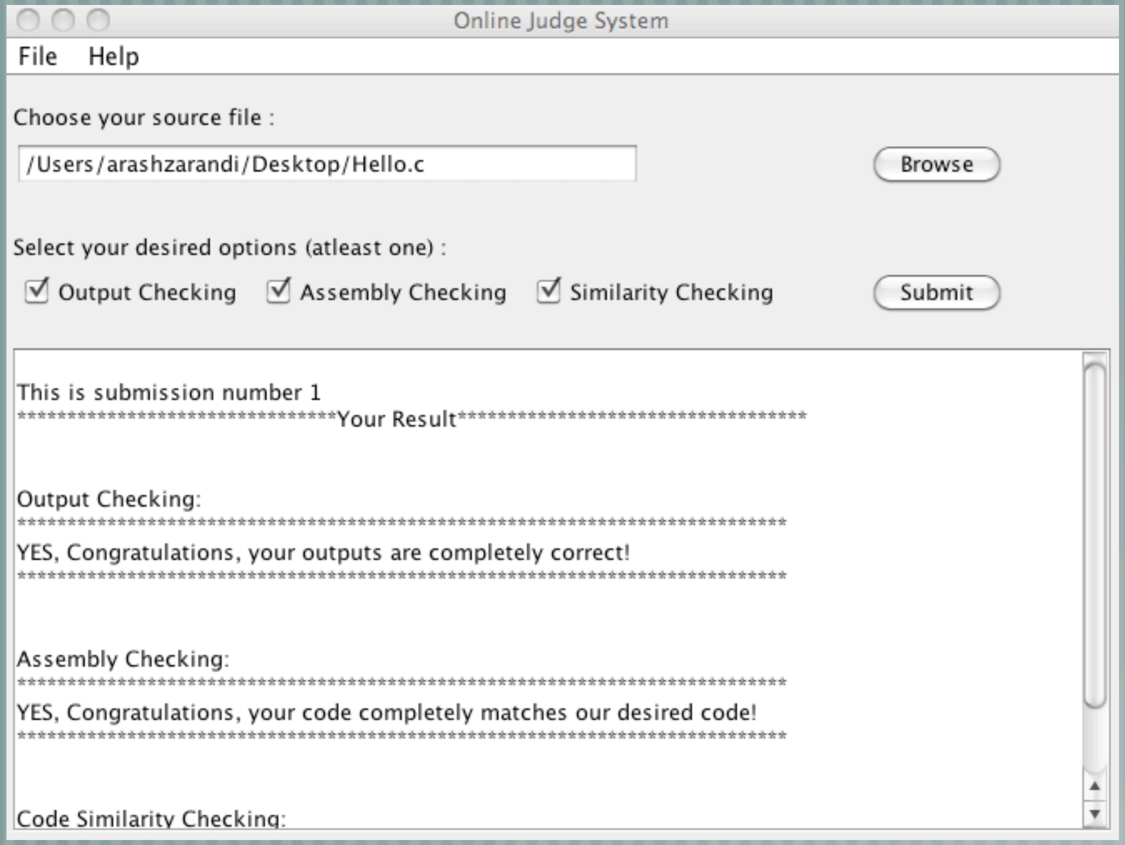
Greedy String Tiling



```
private  HashSet<Match> greedyStringTiling(Token[] array1, Token[] array2){
    HashSet<Match> tiles = new HashSet<Match>();
    int maxMatch = 0;
    do {
        HashSet<Match> matches = new HashSet<Match>();
        for (int i = 0; i < array1.length; i++) {
            for (int k = 0; k < array2.length; k++) {
                int j = 0;
                while((array1[i+j].token == array2[k+j].token) && !array1[i+j].isMarked && !array2[k+j].isMarked)
                    j++;

                if (j == maxMatch)
                    addNonOverlapping(matches, new Match(i, k, j));
                else if (j > maxMatch){
                    matches.clear();
                    matches.add(new Match(i, k, j));
                    maxMatch = j;
                }
            }
        }
        for (Match match : matches) {
            for (int j = 0; j < match.lenght; j++){
                array1[match.a+j].isMarked = true;
                array2[match.b+j].isMarked = true;
            }
            tiles.add(match);
        }
    } while (maxMatch > 0);

    return tiles;
}

private void addNonOverlapping(HashSet<Match> matches, Match match){
    for (Match m : matches) {
        if ((m.a <= match.a && m.a+m.lenght >= match.a) || (m.a <= match.a+match.lenght && m.a+m.lenght >= match.a+match.lenght)
                ||(m.b <= match.b && m.b+m.lenght >= match.b) || (m.b <= match.b+match.lenght && m.b+m.lenght >= match.b+match.lenght))
            return;
    }

    matches.add(match);
}
```

$$sim(A,B) = \frac{2 \cdot coverage(tiles)}{|A| + |B|}$$

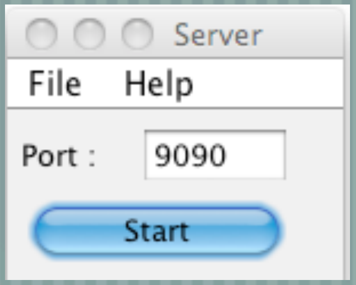
$$coverage(tiles) = \sum_{match(a,b,length) \in tiles} length$$



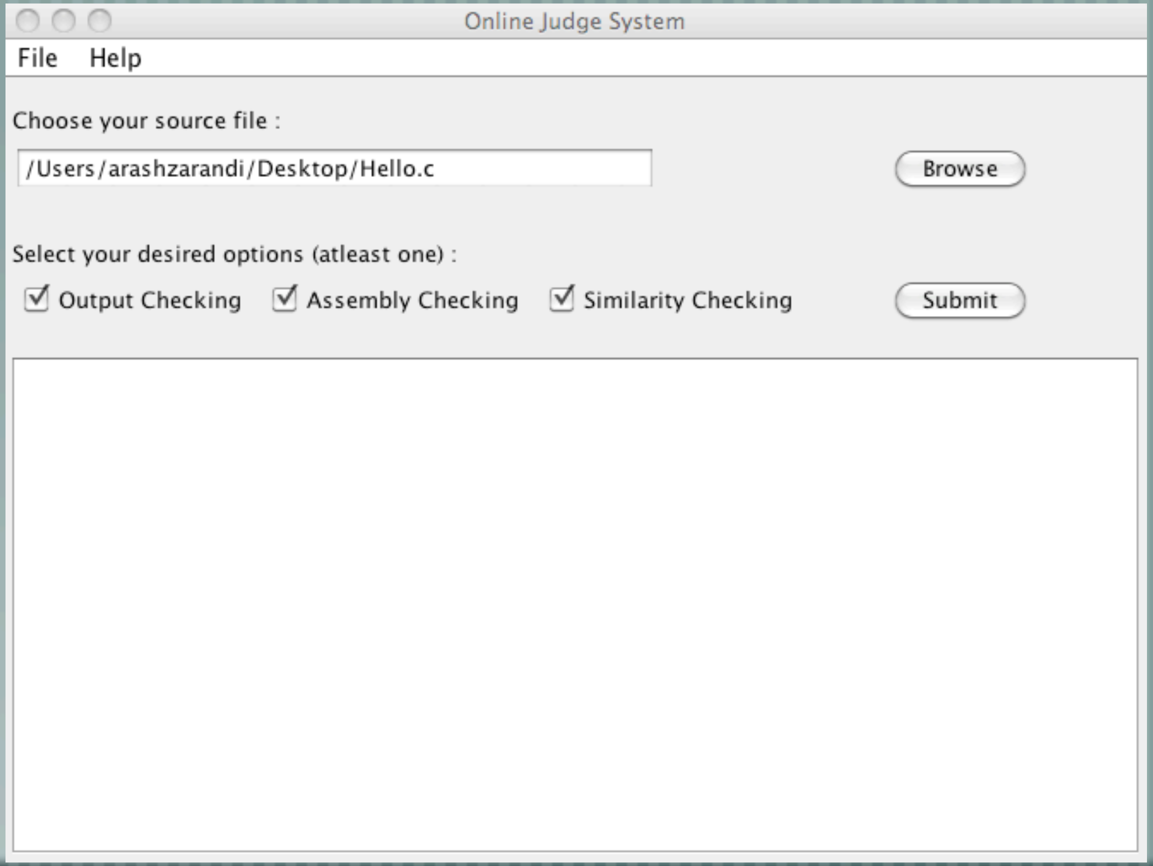
Choose your source file : /Users/arashzarandi/Desktop/Hello.c

Select your desired options (atleast one) :
✓ Output Checking ✓ Assembly Checking ✓ Similarity Checking

```
This is submission number 1
*********************************Your Result************************************


Output Checking:
********************************************************************************
YES, Congratulations, your outputs are completely correct!
********************************************************************************


Assembly Checking:
********************************************************************************
YES, Congratulations, your code completely matches our desired code!
********************************************************************************


Code Similarity Checking:
```



Port : 9090



Choose your source file : /Users/arashzarandi/Desktop/Hello.c

Select your desired options (atleast one) :
✓ Output Checking ✓ Assembly Checking ✓ Similarity Checking

Demo

Any Questions ??

Special Thanks to :
Dr.Ali Hamzeh
Saeed Kazemi
and All Other Who Helped Us Through Out Making This Project.

END