

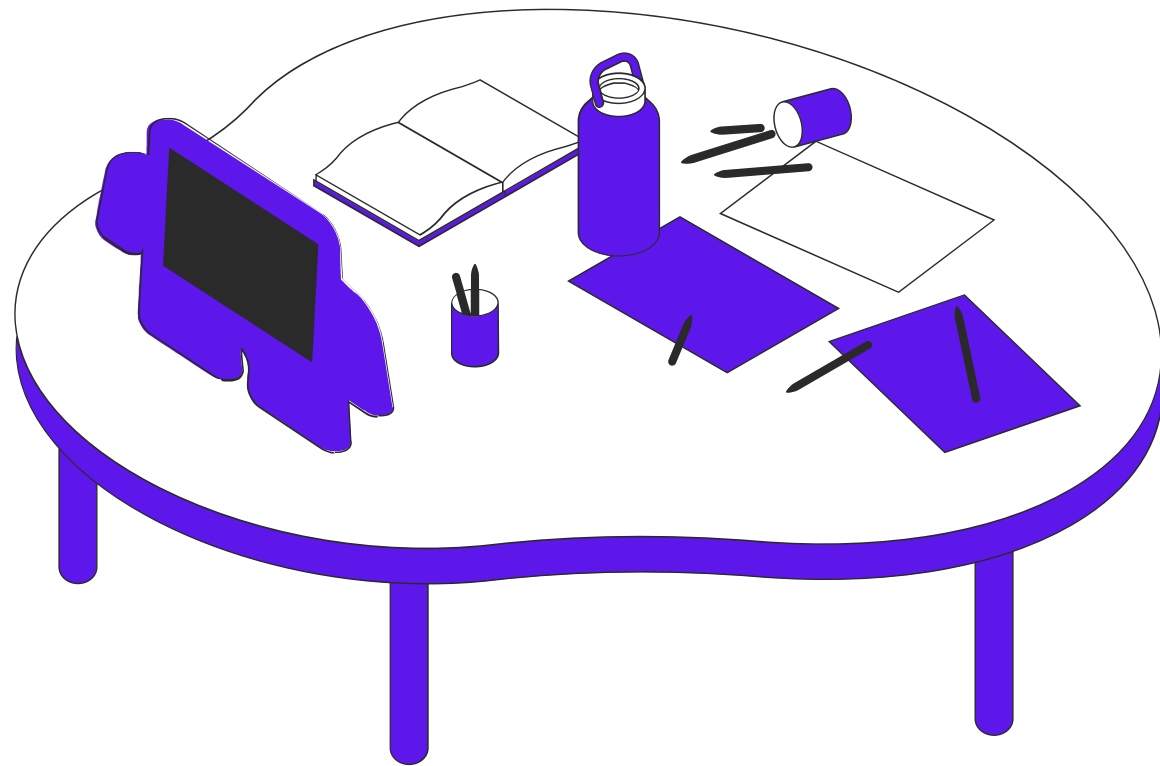
**Graduation Project**

# **EscapeAPP**

**Mustafa Babacan & Tuna Han Öztürk**



# GOAL OF THE PROJECT



The goal of the project is to provide a platform for students and instructors to compare and evaluate solution files.

It aims to automate processes such as comparing student's solutions with instructor's solutions, scoring them, and providing feedback.

This is intended to reduce the time spent on evaluating student performance, provide quick and effective feedback to students, and make the assessment process more efficient.

What are the main technologies used in the project?

- Spring Boot
- JUnit
- Maven
- Java
- Spring MockMVC
- Mockito
- HTML
- CSS
- Javascript



# Spring Boot

Provides a framework for building robust and scalable Java applications, especially web applications.

# JUnit

A popular unit testing framework  
for Java programming language.

The JUnit logo features the word "JUnit" in a serif font. The "J" is green, and the "Unit" is red. The letters are bold and have a classic, slightly ornate design.

# Test Scripts

**FileUploadControllerTest:** This is a JUnit test class for the handleFileUpload method in the FileUploadController class. The handleFileUpload method is responsible for handling file uploads, extracting content from ZIP files, and comparing student and instructor files.

**ZipExtractorServiceTest:** This is a JUnit test class for the ZipExtractorService class, and it primarily focuses on testing the functionality of the extractAndCompareFiles and getInstructorFolderPath methods.

**ZipFileEntriesTest:** This JUnit test class, ZipFileEntriesTest, is designed to test the functionality of the ZipFileEntries class.

# Important Libraries / Dependencies

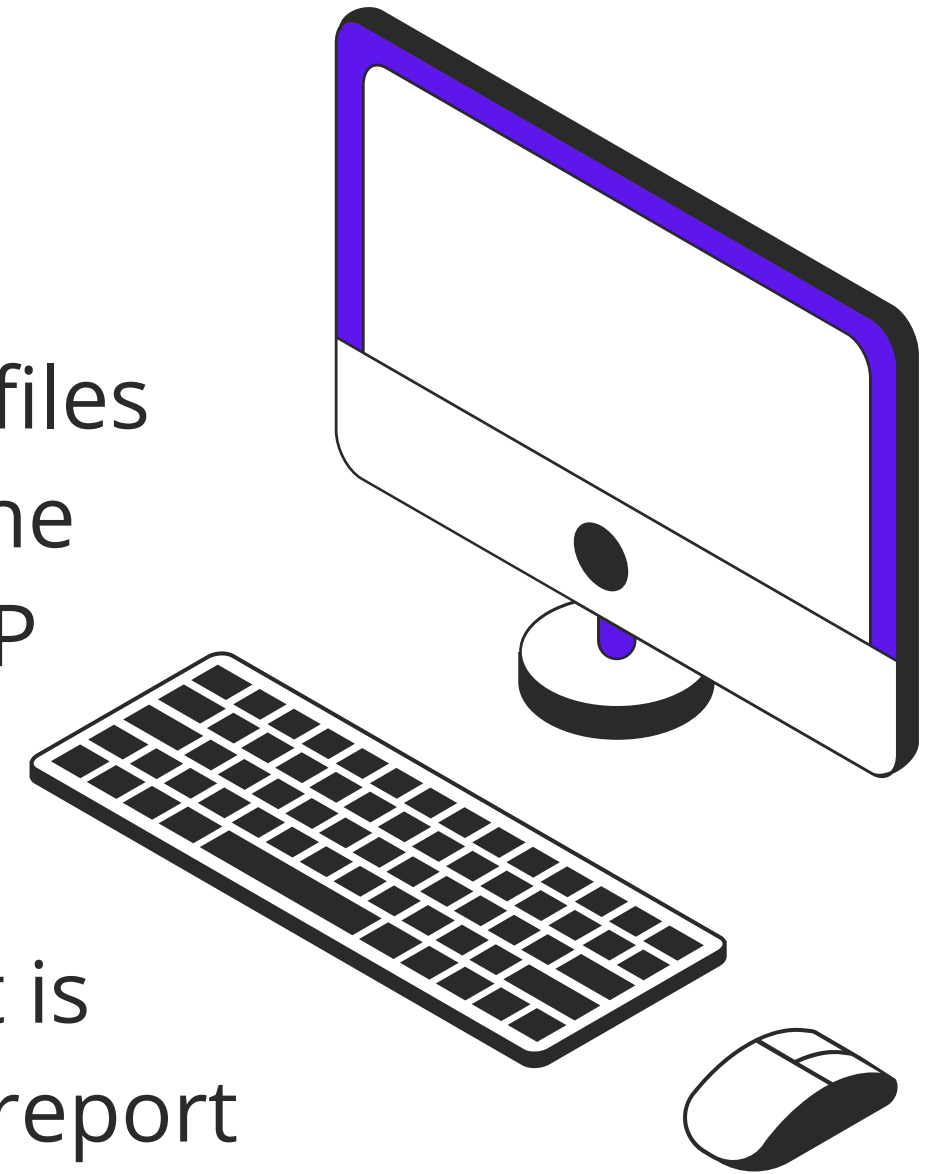


- Spring Boot Starter WEB
- Spring Boot DevTools
- Spring Mock
- Spring Boot Test Starter

# How The Project Works

**File Upload and Number of Questions:** Users can upload zip files containing student and instructor solutions. They can choose the number of questions. These files are sent to the server via HTTP requests.

**Report Generation:** After all questions are compared, a report is created in the project hierarchy for each student solution. The report includes details such as whether each question was passed or not and the total score based on the number of questions passed.





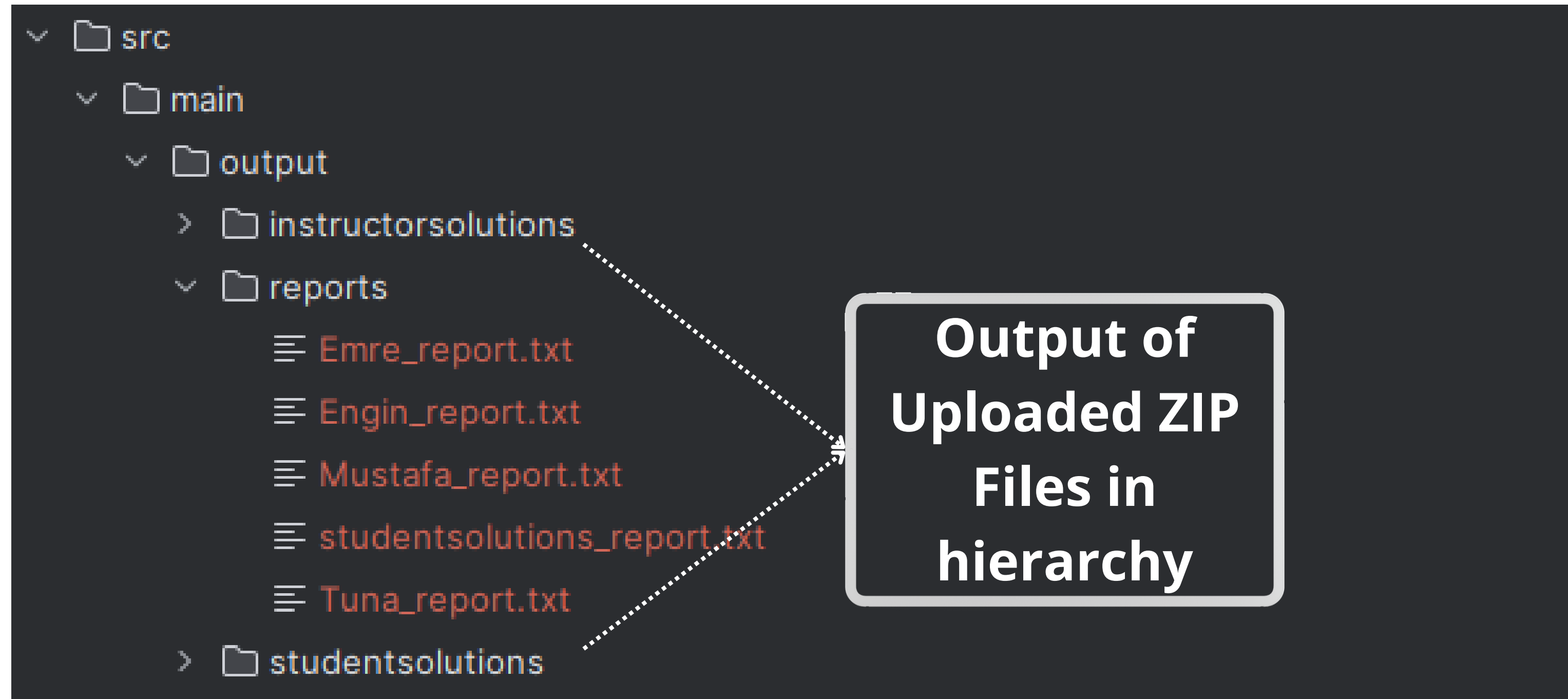
# Upload ZIP Files

Enter the number of questions:

Upload Student ZIP File:

Upload Instructor ZIP File:

*Upload ZIP Files Page*



*Hierarchy After Upload*

≡ Tuna_report.txt ×	
1	Student: Tuna
2	Question 1: Passed
3	Question 2: Passed
4	Question 3: Passed
5	Total Score: 30
6	
7	

≡ Emre_report.txt ×	
1	Student: Emre
2	Question 1: Failed
3	Question 2: Passed
4	Question 3: Failed
5	Total Score: 10
6	
7	

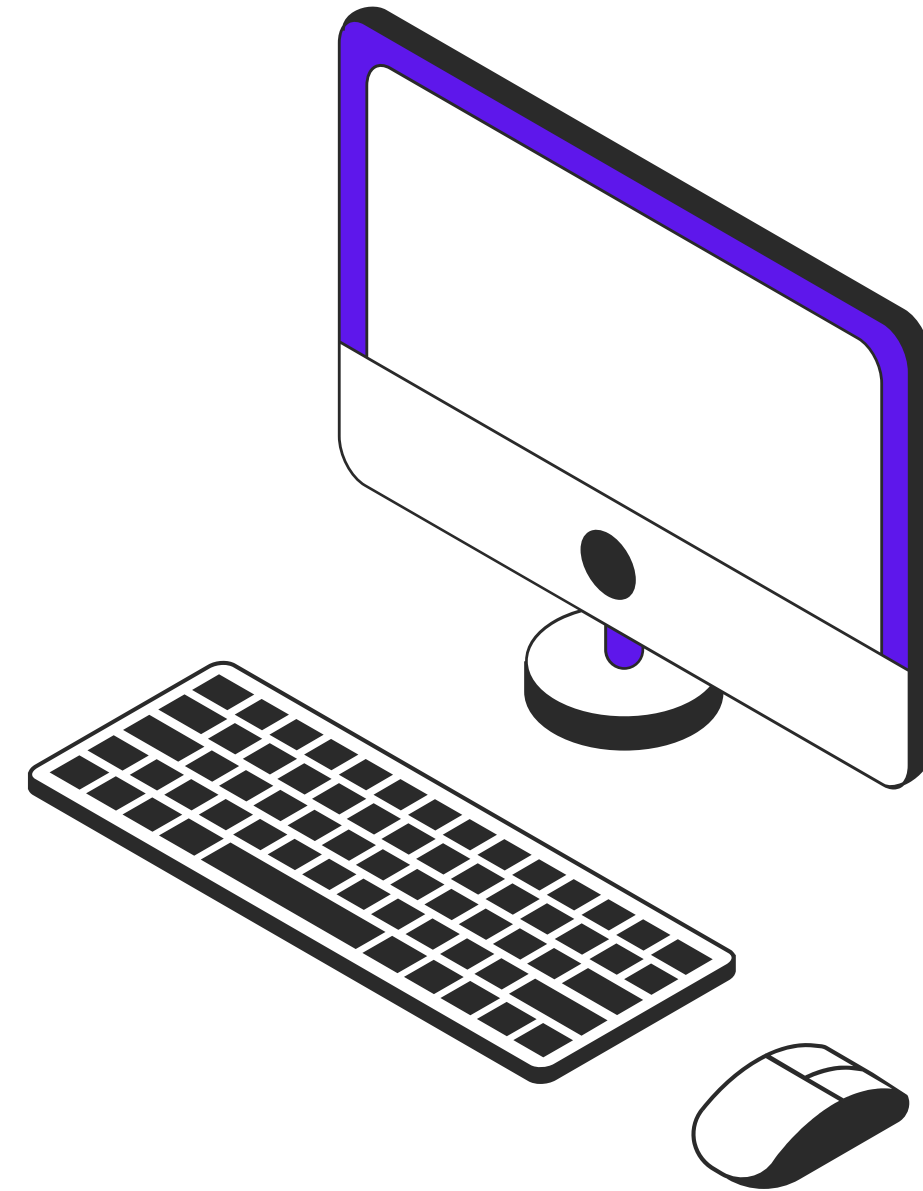
## *Report Content Examples*

# How does it work in the background?

**File Handling:** The FileUploadController class handles the file upload process. It receives the uploaded zip files, along with other necessary parameters such as the number of questions, via POST requests.

**Zip Extraction:** The uploaded zip files are passed to the ZipExtractorService, which extracts the contents of the zip files. It then compares the extracted files between student and instructor solutions.

**Comparison:** The extracted files from the student and instructor zip files are compared. For each question, the content of the corresponding files (e.g., question1.java) is compared. If the content matches, the question is considered passed; otherwise, it is considered failed.



# Things to pay attention

## ZIP Files and Creation Order

Each "students.zip file" should contain folders consisting of student names, and each student folder should contain a question folder within itself. The java extension file of the relevant question should be in these question folders.

for example;

studentsolutions.zip>studentsolutions (folder)>

Mustafa (folder)>Q1 (folder)>question1.java

1

studentsolutions.zip - ZIP arşivi, açılmış boyut 619 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
studentsolutions	619	597	Dosya klasörü	23.01.2024 18:07		

2

studentsolutions.zip\studentsolutions - ZIP arşivi, açılmış boyut 619 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
Tuna	157	144	Dosya klasörü	22.01.2024 23:48		
Mustafa	157	144	Dosya klasörü	22.01.2024 23:48		
Engin	135	136	Dosya klasörü	23.01.2024 18:07		
Emre	170	173	Dosya klasörü	23.01.2024 18:08		

3

studentsolutions.zip\studentsolutions\Mustafa - ZIP arşivi, açılmış boyut 619 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
Q1	0	0	Dosya klasörü	22.01.2024 23:49		
Q2	37	39	Dosya klasörü	22.01.2024 23:49		
Q3	120	105	Dosya klasörü	22.01.2024 23:49		

4

studentsolutions.zip\studentsolutions\Mustafa\Q3 - ZIP arşivi, açılmış boyut 619 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
question3.java	120	105	IntelliJ IDEA	25.12.2023 12:44	6C94178D	

# Things to pay attention

Each "instructorsolutions.zip file" should contain folders consisting of questions, and the java extension file of the relevant question should be in these question folders.

for example;

```
instructorsolutions.zip>instructorsolutions  
(folder)>Q1 (folder)>question1.java
```

1

↑ instructorsolutions.zip - ZIP arşivi, açılmış boyut 236 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
instructorsolutio...	236	223	Dosya klasörü	23.01.2024 18:11		

2

↑ instructorsolutions.zip\instructorsolutions - ZIP arşivi, açılmış boyut 236 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
Q1	0	0	Dosya klasörü	22.01.2024 23:47		
Q2	37	39	Dosya klasörü	22.01.2024 23:47		
Q3	120	105	Dosya klasörü	22.01.2024 23:47		
Q4	79	79	Dosya klasörü	23.01.2024 18:12		

3

↑ instructorsolutions.zip\instructorsolutions\Q1 - ZIP arşivi, açılmış boyut 236 bayt						
Ad	Boyut	Sıkı. boyut	Tür	Değişme	CRC32	
..			Dosya klasörü			
question1.java	0	0	IntelliJ IDEA	25.12.2023 12:52	00000000	



**Thank you for  
listening to us.**

