1 Write a program to set up or configure REST assured.

```
package in.assertions.RestAssuredDemo;


import org.testng.annotations.Test;


import io.restassured.RestAssured;


public class GetBearerTokenDemo{
	@Test
	public void createResources() {
		RestAssured
		.given()
		.contentType("application/json")
		.header("Authorization"," Bearer 1498d64025e554a3e2c77dbb9ba48fbdb9cf2f97635f2b4036ee373513eb2ea9")
		.when()
		.get("https://gorest.co.in/public/v2/users/4495127")
		.then()
		.statusCode(200)
		.log().all();
	}
}
```

2 Write a program to demonstrate automation of GET request and response.

```
package in.assertions.RestAssuredDemo;


import org.testng.annotations.Test;


import io.restassured.RestAssured;


public class GETDemo {
	@Test
```

```
    public void getServerResponse() {
        RestAssured
        .when()
        .get("https://restcountries.com/v3.1/all")
        .then()
        .assertThat()
        .statusCode(200)
        .log().all();
    }

}
```

3 Write a program to demonstrate automation of POST request and response.

```
package in.assertions.RestAssuredDemo;

import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import java.util.HashMap;

public class POSTDemo {
    HashMap<String, String> map =new HashMap<String,String>();

    @BeforeMethod
    public void createPayLoad() {
        map.put("name","morpheus");
        map.put("job","painter");
        RestAssured.baseURI="https://reqres.in/";
        RestAssured.basePath="/api/users";

    }
```

```
	@Test
	public void createResources() {
		RestAssured
		.given()
			.contentType("application/json")
			.body(map)
		.when()
			.post()
		.then()
			.statusCode(201)
			.log().all();
	}
}
```

4 Write a program to send XML payload in REST assured, parse XML response in REST assured, parse JSON response in REST assured, and explore native logging of REST assured.

Xml:

```
package in.assertions.RestAssuredDemo;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;



import org.apache.commons.io.IOUtils;

import org.testng.annotations.Test;



import io.restassured.RestAssured;



public class SOAPDemo {

	@Test
```

```
            public void getCapital() throws IOException {

                        FileInputStream fis= new FileInputStream(new File("capitals.xml"));


            RestAssured.baseURI="http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL";

                        RestAssured

                        .given()

                                    .header("content-Type","text/xml")

                                    .body(IOUtils.toString(fis,"UTF-8"))

                        .when()

                                    .post()

                        .then()

                                    .statusCode(200)

                                    .log().all();


            }


            }
```

Json:

```
 package in.assertions.RestAssuredDemo;


import java.util.HashMap;


import org.junit.Assert;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;

public class PutDemoBearerToken {
    HashMap<String, String> map =new HashMap<String,String>();
    @BeforeMethod
    public void createPayLoad() {
        map.put("name", "spiderman");
        map.put("email", "spider2@gmail.com");
        map.put("gender", "male");
        map.put("status", "active");
        RestAssured.baseURI="https://gorest.co.in";
        RestAssured.basePath="public/v2/users/4495127";
    }


    @Test
    public void updateResources() {
        Response response=RestAssured
        .given()
        .contentType("application/json")
        .body(map)
        .header("Authorization"," Bearer 1498d64025e554a3e2c77dbb9ba48fbdb9cf2f97635f2b4036ee373513eb2ea9")
```

```
                .when()

                .put()

                .then()

                .extract().response();



                JsonPath jsonpath =response.jsonPath();

                Assert.assertTrue(jsonpath.get("name").toString().equals("spiderman"));

        }



}
```

5 Write a program to demonstrate GET, POST, XML, and JSON in REST Assured.

```
package in.assertions.RestAssuredDemo;

import java.util.HashMap;
import java.util.UUID;

import org.junit.Assert;



import org.testng.annotations.Test;
import static org.hamcrest.Matchers.~~is~~;

import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;




public class Create_Update_Delete_Resource {
        HashMap<String, String> map =new HashMap<String,String>();

        UUID uuid =UUID.randomUUID();
        int id;
        Response response;
        JsonPath jsonpath;
        @Test(priority=0)

        public void createPayLoad() {
                map.put("name", "Captain Planet");
                map.put("email", uuid + "@gmail.com");
                map.put("gender", "male");
                map.put("status", "active");
                RestAssured.baseURI="https://gorest.co.in";
```

```
            RestAssured.basePath="public/v2/users";
      }
      @Test(priority=1)

      public void createResource() {
            response=RestAssured
            .given()
            .contentType("application/json")
            .body(map)
            .header("Authorization"," Bearer 
1498d64025e554a3e2c77dbb9ba48fbdb9cf2f97635f2b4036ee373513eb2ea9")
            .when()
            .post()
            .then()
            .log().all()
            .extract().response();

            jsonpath =response.jsonPath();
            id =jsonpath.get("id");
            System.out.println(id);
      }
      @Test(priority=2)

      public void verifyResource() {
            System.out.println("Id= "+ id);
            RestAssured
            .given()
            .contentType("application/json")
            .header("Authorization"," Bearer 
1498d64025e554a3e2c77dbb9ba48fbdb9cf2f97635f2b4036ee373513eb2ea9")
            .when()
            .get("https://gorest.co.in/public/v2/users/"+id)
            .then()
            .statusCode(200)
            .log().all()
            .extract().response();


      Assert.assertTrue(jsonpath.get("name").toString().equals("Captain 
Planet"));
      }
      @Test(priority=3)

      public void updateResource() {
            map.put("name", "Captain Planett");
            map.put("email", uuid + "@gmail.com");
            map.put("gender", "male");
            map.put("status", "active");
            RestAssured.baseURI="https://gorest.co.in";
            RestAssured.basePath="public/v2/users/" + id;

            RestAssured
            .given()
            .contentType("application/json")
            .header("Authorization"," Bearer 
1498d64025e554a3e2c77dbb9ba48fbdb9cf2f97635f2b4036ee373513eb2ea9")
            .body(map)
            .when()
            .put()
            .then()
            .statusCode(200)
            .log().all()
            .assertThat()
            .body("name", is("Captain Planett"));
    }
    @Test(priority=4)

    public void deleteResource() {
            RestAssured.baseURI="https://gorest.co.in";
            RestAssured.basePath="public/v2/users/" + id;
            RestAssured
            .given()
            .contentType("application/json")
            .header("Authorization"," Bearer
1498d64025e554a3e2c77dbb9ba48fbdb9cf2f97635f2b4036ee373513eb2ea9")
            .when()
            .delete()
            .then()
            .statusCode(204);

}}
```

6 Write a program to demonstrate OAuth authorization in REST API using REST assured.

```
package in.assertions.RestAssuredDemo;



import org.testng.annotations.Test;



import io.restassured.RestAssured;



public class APIKeyDemo {

    @Test

    public void getWheatherInfo() {

            RestAssured

            .given()

            .param("q", "chennai")

            .param("appid","99bd468f9c77a835c2aac178f125f6ed")

            .when()

            .get("https://api.openweathermap.org/data/2.5/weather")

            .then()

            .statusCode(200)
            .log().all();

      }



}
```

8 Write a program to configure Log4j in Eclipse Maven project.

```
<properties>
            <serenity.version>1.5.8</serenity.version>
            <serenity.maven.version>1.5.8</serenity.maven.version>
            <slf4j.version>1.6.1</slf4j.version>

      <maven.failsafe.plugin.version>2.18</maven.failsafe.plugin.version>

      <maven.compiler.plugin.version>3.2</maven.compiler.plugin.version>
            <java.version>1.8</java.version>
      </properties>

      <dependencies>
            <dependency>
                  <groupId>net.serenity-bdd</groupId>
                  <artifactId>serenity-core</artifactId>
                  <version>${serenity.version}</version>
            </dependency>
            <dependency>
                  <groupId>net.serenity-bdd</groupId>
                  <artifactId>serenity-rest-assured</artifactId>
                  <version>${serenity.version}</version>
            </dependency>
            <!-- https://mvnrepository.com/artifact/log4j/log4j -->
            <dependency>
                <groupId>log4j</groupId>
                <artifactId>log4j</artifactId>
                <version>1.2.17</version>
            </dependency>
            <dependency>
                  <groupId>org.slf4j</groupId>
                  <artifactId>slf4j-simple</artifactId>
                  <version>${slf4j.version}</version>
            </dependency>
            <!-- https://mvnrepository.com/artifact/org.testng/testng -->
            <dependency>
                  <groupId>org.testng</groupId>
                  <artifactId>testng</artifactId>
                  <version>7.1.0</version>
                  <scope>test</scope>
            </dependency>

            <!-- API, java.xml.bind module -->
            <dependency>
                  <groupId>jakarta.xml.bind</groupId>
                  <artifactId>jakarta.xml.bind-api</artifactId>
                  <version>2.3.2</version>
            </dependency>

            <!-- Runtime, com.sun.xml.bind module -->
```

```
		<dependency>
			<groupId>org.glassfish.jaxb</groupId>
			<artifactId>jaxb-runtime</artifactId>
			<version>2.3.2</version>
		</dependency>


		<dependency>
			<groupId>in.Amazon</groupId>
			<artifactId>Amazon</artifactId>
			<version>0.0.1-SNAPSHOT</version>
		</dependency>
	</dependencies>
```

Class code:

```
package in.assertions.RestAssuredDemo;

import org.apache.log4j.Level;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.testng.annotations.BeforeClass;



import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;

public class TestBase {
	public static RequestSpecification httprequest;
	public static Response response;

	public Logger logger;

	@BeforeClass
	public void setUp() {
		logger =Logger.getLogger("logDemo");
		PropertyConfigurator.configure("log4j.properties");
		logger.setLevel(Level.DEBUG);
	}

}
```

9 Write a program to generate logs on Eclipse Console using Log4j.

```
package in.assertions.RestAssuredDemo;

import org.apache.log4j.Level;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.testng.annotations.BeforeClass;



import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;

public class TestBase {
      public static RequestSpecification httprequest;
      public static Response response;

      public Logger logger;

      @BeforeClass
      public void setUp() {
            logger =Logger.getLogger("logDemo");
            PropertyConfigurator.configure("log4j.properties");
            logger.setLevel(Level.DEBUG);
      }

}
```

10 Write a program to write log file using Log4j.

```
og4j.rootLogger=ERROR,stdout
log4j.logger.com.endeca=INFO
log4j.logger.LogDemo=DEBUG, dest1
# Logger for crawl metrics
log4j.logger.com.endeca.itl.web.metrics=INFO

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%p\t%d{ISO8601}\t%r\t%c\t[%t]\t%m%n
```