# Java Developer test assignment

## Introduction

You are provided with *xml* file containing a representation of files hierarchy.

Example:

```
<node is-file="false">
  <name>/</name>
  <children>
    <child is-file="true">
      <name>file-776194140.xml</name>
    </child>
    <child is-file="false">
      <name>dir-880176375</name>
      <children>
        <child is-file="true">
          <name>file-1073842118.java</name>
        </child>
        <child is-file="false">
          <name>dir-2145307015</name>
          <children>
            <child is-file="true">
              <name>file-1498940214.xhtml</name>
            </child>
          </children>
        </child>
      </children>
    </child>
  </children>
</node>
```

One node may be either file or directory, which is defined by attribute *is-file.*

## What to do?

Implement command-line application which will output all full paths for the given input.

```
$ java -jar assignment.jar -f <xml_file> -s <input>
```

where `<xml_file>` is a path to the provided xml file; `<input>` is search string to filter paths with.

## Examples

No search input:

```
$ java -jar assignment.jar -f test-files.xml
```

```
/file-776194140.xml
/dir-880176375/file-1073842118.java
/dir-880176375/dir-2145307015/file-1498940214.xhtml
```

Exact search input:

```
$ java -jar assignment.jar -f test-files.xml -s file-1498940214.xhtml
```

```
/dir-880176375/dir-2145307015/file-1498940214.xhtml
```

Simple search input:

```
$ java -jar assignment.jar -f test-files.xml -s ‘*.java’
```

```
/dir-880176375/file-1073842118.java
```

Extended search input:

```
$ java -jar assignment.jar -f test-files.xml -S ‘.*?[a-z]{4}-\\d+\.[a-z]+’
```

```
/file-776194140.xml
/dir-880176375/file-1073842118.java
/dir-880176375/dir-2145307015/file-1498940214.xhtml
```

## Compilation and test environment

Prepare Maven project. It is allowed to use any libraries you might find useful; Java SE 8+ or Kotlin.

Build should run without problems with Maven 3.5.0.