Take-Home Programming Test (url-engine)

Disclaimer

The following test is solely property of Palo Alto Networks.
You accept not to disclose any part of the questions and answers developed for the test.

This program is to be finished in **72 hours** by the candidate alone.
Use of Google or other generic research is allowed.
State clearly any assumptions made and their limitations.

We expect a fully working C or C++ program that meets the requirements.
We may compile, run and test the submitted source code.

The Test

Design and develop a program "`url-engine`".

Environment

OS: Linux, e.g. Ubuntu 18.04
Language: C or C++

Input & Output

Command line:

```
url-engine <algorithm(posix|self)> configuration.xml urls.txt
```

Example `configuration.xml`:

```
<config>
  <set id="1">
    <pattern>*.aaa.com</pattern>
    <pattern>*.bb.cc/*</pattern>
    <pattern>www.*.com</pattern>
  </set>
  <set id="2">
    <pattern>www.*.com</pattern>
  </set>
</config>
```

In each pattern:

1) The first "/" is a delimiter between the hostname and pathname
2) "*" is a wildcard that matches differently before and after the delimiter
    a) On the hostname, before the first "/", it matches any characters that is not "/"
    b) On the pathname, after the first "/", it matches any characters

Example `urls.txt`:

```
www.aaa.com
aa.bb.cc/ddx/eee
www.abc.com
xx.bb.cc/ddd
...
url1000000.com/path999
```

Requirements

1. `url-engine` parses and loads the patterns from `configuration.xml`. Use APIs from `libxml2`. For C++ solutions, OK to design your own class to wrap `libxml2`, but don't use existing 3rd party wrapper libraries.

2. `url-engine` reads `urls.txt` and matches the URLs against the patterns loaded in step 1. Print out each URL along with its matching pattern set number and pattern to `STDOUT`. If multiple patterns are matched, then print all of them.

Example output:

```
URL:www.aaa.com;Set:1:Pattern:*.aaa.com;Set:1:Pattern:www.*.com;Set
:2:Pattern:www.*.com
URL:aa.bb.cc/ddx/eee;Set:1:Pattern:*.bb.cc/*
URL:www.abc.com;Set:1:Pattern:www.*.com;Set:2:Pattern:www.*.com
URL:xx.bb.cc/ddd;Set:1:Pattern:*.bb.cc/*
...
URL:url1000000.com/path999
```

3. You should implement two matching algorithms:
    1) "`posix`" by POSIX regular expression library.
    2) "`self`" by your own implementation

Selection of algorithms can be specified by an argument to `url-engine`.

4. `url-engine` should scale to support large input data.
`urls.txt` could contain millions of URLs. Memory is not enough to buffer the entire file.
`configuration.xml` could contain hundreds of pattern sets and in each set there could be hundreds of patterns.