

CMPS 258 – PROGRAMMING LANGUAGES – SPRING 2021
WEEK 2 ASSIGNMENT

In this assignment, you will write your own datatype and use case expressions and pattern matching to traverse it. The datatype you will write represents the directory structure of a file system in a computer.

1. Write a datatype binding for a file system object `FSObject` that can be one of the following:
 - A file which has two fields: a name (stored as a `string`) and the size of the file (stored as an `int`)
 - A directory which has two fields: a name (stored as a `string`) and a list of the file system objects the directory contains
 - A link which has two fields: a name (stored as a `string`) and the path to the file or directory that it links to (stored as a `string`)
2. Create a value `myFS` of type `FSObject` that stores a file system object with the following hierarchy:

```
dirA
+---- dirB
|      +---- dirC
|      |      +---- file1 (4096 B)
|      |      +---- file2 (2097152 B)
|      |      +---- linkX -> dirA/dirD/file4
|      |      +---- linkY -> dirA/dirD/file4
+---- dirD
      +---- file3 (4194304 B)
      +---- file4 (128 B)
      +---- linkZ -> dirA/dirB/dirC/file1
```

3. Write a function `totalSize` that takes a value of type `FSObject` and returns the total size of all the files in the hierarchy.
4. Write a function `containsLinks` that takes a value of type `FSObject` and returns `true` if the hierarchy contains any links, and `false` otherwise.
5. Write a function `getFilesLargerThan` that takes a value of type `FSObject` and an integer `n` and returns a `string list` with the names of all the files larger than `n`. It is okay to use the operator `@` for concatenating two lists (e.g., `xs @ ys` concatenates the two lists `xs` and `ys`).
6. Write a function `countLinksTo` that takes a value of type `FSObject` and a string `p` and returns the number of links that point to the path `p`.

Evaluating a correct homework solution should generate the bindings below. However, keep in mind that generating these bindings does not guarantee that your solutions are correct. Make sure to test your functions before submitting.

```
datatype FSObject = <omitted>
val myFS = <omitted>
val totalSize = fn : FSObject -> int
val containsLinks = fn : FSObject -> bool
val getFilesLargerThan = fn : FSObject * int -> string list
val countLinksTo = fn : FSObject * string -> int
```

Assessment

Solutions should be:

- Correct
- In good style, including indentation and line breaks
- Written using features discussed in class

Submission Instructions

Put all your solutions in one SML file and submit it via Moodle. The file should be named "<id>.sml" where <id> is your AUBnet ID (e.g., abc01.sml). Do not submit any other files or compressed folders.