

Principles of Programming, Spring 2006
Practice 10
The nML Module System

Daejun Park, Heejong Lee
Programming Research Lab.@SNU

May 25, 2006

Consider the following code.

```
structure Mapping =  
struct  
  
  exception NotFound  
  
  val create = []  
  
  fun lookup d [] = raise NotFound  
    | lookup d ((e,r) :: es) =  
      if d = e  
      then  
        r  
      else  
        lookup d es  
  
  fun insert d r [] = [(d, r)]  
    | insert d r ((e,s) :: es) =  
      if d = e  
      then  
        (d,r) :: es  
      else  
        (e,s) :: (insert d r es)  
  
end  
  
signature SIMAPPING =  
sig  
  exception NotFound
```

```

    val create : (string * int) list
    val lookup : string -> (string * int) list -> int
    val insert : string -> int -> (string * int) list ->
        (string * int) list
end

```

```

structure SiMapping : SIMAPPING = Mapping

```

```

signature SSIMAPPING =
sig
  exception NotFound
  val lookup : string -> (string * int) list -> int
end

```

```

structure SSiMapping : SSIMAPPING = Mapping

```

What is the result of the following expressions?

```

val em = SiMapping.create
val m = SiMapping.insert "a" 1 (SiMapping.insert "b" 2 em)
SiMapping.lookup "b" m
SSiMapping.lookup "b" m

```

```

val em' = Mapping.create
val m' = Mapping.insert 1 10 (Mapping.insert 2 20 em')
Mapping.lookup 1 m'

```

Find the BUGs in the following.

```

val em'' = Mapping.create
val m'' = Mapping.insert "a" 1 (Mapping.insert 2 20 em'')
Mapping.lookup "a" m''

```

Define a structure `CMapping` and a signature `CMAPPING`, which can recognize both integer and string as the key of the mapping.