```
allRolls choices (vs. n) = case pop choices vs of
   Nothing -> [ ([], n-1) ]
   Just ((chosen. v). (choices. vs)) ->
      allRolls choices (vs, n-1) >>=

          \(roll. ) -> [ (d:roll,  n-1)
             | d <- rollList ]
          where
            rollList = if chosen then [v] else [ 1..6 ]
```

```
allRolls choices (vs. n) = case pop choices vs of
   Nothing -> [ ([], n-1) ]
   Just ((chosen. v). (choices. vs)) ->
      allRolls choices (vs, error "Didn't expect
        to use") >>=
          \(roll. ) -> [ (d:roll,  n-1)
             | d <- rollList ]
          where
            rollList = if chosen then [v] else [ 1..6 ]
```