

```

01 allRolls :: (DiceChoice, DiceVals)
02         -> Integer
03         -> [ (DiceVals, Integer) ]
04 allRolls t n = case pop t of
05     Nothing -> [ ([], n-1) ]
06
07     Just ((chosen, v), t) ->
08         allRolls t (error "Didn't expect to use")
09         >=> \ (roll, ) -> [ (d:roll, n-1)
10             | d <- rollList ]
11     where
12         rollList = if chosen then [v] else [ 1..6 ]

```

```

01 allRolls :: (DiceChoice, DiceVals)
02         -> Integer
03         -> [ (DiceVals, Integer) ]
04 allRolls t n = [ (vals, n-1) | vals <- allRollsNoN t ]
05
06 allRollsNoN :: (DiceChoice, DiceVals) -> [ DiceVals ]
07 allRollsNoN t = case pop t of
08     Nothing -> [ [] ]
09     Just ((chosen, v), t) ->
10         allRollsNoN t >=>
11         \roll -> [ d:roll | d <- rollList ]
12
13     where
14         rollList = if chosen then [v] else [ 1..6 ]

```