```
lotus Solver :: [Int] -> [Int]
lotus Solver A = recurse (A 0 1)



addToList :: [Int] -> Int -> Int -> [Int]
addToList A index value
    | [A!! index] /= 0 = A
    | otherwise = buildList (A -1 index value)



buildList :: [Int] -> Int -> Int -> Int -> [Int]
buildList A loopIndex index value
    | loopIndex == -1 = buildList (A 0 index value)
    | loopIndex == index = buildList(A loopindex+1 index val): value   <- Prepend instead
    | loopIndex == 49 = []
    | otherwise = buildList(A loopIndex+1 index value): A!! loopIndex
      where B = take index-1 A
```