Version 1:

1. Get input from user
   1. Two lines, first is number, second is string
2. Find secret encoded message inside
   1. Write it out to a file
3. Continue until input is 0

Version 2:

1. Main
   1. Takes in all numbers input and string inputs from file until number is 0, send those to *decrypt*() <- *final*
   2. For each case, send *final* to file
2. getContent()
   1. takes in all values from a given file and creates a list consisting of them
3. decrypt(columns, encodedString)
   1. define two separators that are able to step through the string in what equates to be bostrophudonic order. Each pass through adds the appropriate characters, then updates both separators and other temporary variables as needed. Return the final string as *finalString*

Version 3:

1. main
   1. open file and get all lines of content <- *content*
   2. for numbers in range[0,length of content) <- *x*
      1. *content*[*x*] <- *tempInt*
      2. *content*[*x*+1] <- *tempString*
      3. *decrypt*(*tempInt*, *tempString*) <- *decodedString*
      4. *outToFinal*(*decodedString*)
      5. increment *x*
2. getContent()
   1. open file
   2. get all lines of file <- *content*
   3. strip whitespace from lines
   4. return *content*
3. decrypt(columns, encodedString)
   1. columns\*2-1 <- sep1, originalSep1
   2. 1 <- sep2
   3. False <- done
   4. “” <- finalString
   5. 0 <- x
   6. 0 < - currentIndex
   7. length of encodedString <- length
   8. while not done
      1. x=0
      2. While(x < length)
         1. X=currentIndex
         2. Add encodedString[x] to fullString
         3. X+=sep1
         4. If x<length
            1. Add encodedString[x] to fullString
         5. X+=sep2
         6. If x<length
            1. Add encodedString[x] to fullString
         7. Sep1-=2
         8. Sep2-=2
      3. currentIndex++
      4. if currentIndex==originalSep1
         1. done=true
   9. return finalString
4. outToFinal(decodedString)
   1. print(decodedString)