\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

V. encodeMethod1 (inFile, encodeFile) // encode zero’s and NO wrapped around

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// The algorithm steps may contain bugs, you are responsible to debug it.

Step 0:

numRows, numCols, minVal, maxVal 🡨 get from image header

encodeFile 🡨 output numRows, numCols, minVal, maxVal to encodeFile

row 🡨 0

Step 1: col 🡨 0

length🡨 0

currVal 🡨 read the next pixel (integer),from inFile \*one integer at a time\*

Step 2: col++

Step 3: nextVal 🡨 read the next pixel (integer), one pixel at a time, from inFile

Step 4: if nextVal == currVal

length ++

else

encodeFile 🡨 output length

currVal 🡨 nextVal

length 🡨 1

encodeFile 🡨 output row and col and currVal to encodeFile

Step 5: repeat step 2 to step 4 while length < numCols

Step 6: row ++

Step 7: output length

Step 8: repeat Step 1 to Step 7 until end of file or while row < numRows

// You may need to output the length for the last run after step 8.

step 9: closed all files