Student: Michael Grossman

Due Date: 5/19/2022

Algorithm Steps for RLE with 0's and no wrap-around given a file representing an image with a header:

1. InputFile ←args[1];
2. OutputFile ←args[2];
3. numRows, numCols, minVal, maxVal ← inputFile
4. outputFile ←numRow, numCols, minVal, maxVal
5. row ← 0
6. col ← 0
7. currVal ← inputFile
8. length ←1
9. startR ← row
10. startC ← col
11. val ← currentVal
12. outputFile ← startR, startC, val
13. col++
14. nextVal ← inputFile
15. if nextVal == currVal:
16. length++;
17. else:
18. outputFile ← length
19. currVal ← nextVal
20. length ← 1
21. startR ← row
22. startC ← col
23. val ← currentVal
24. outputFile ←startR, startC, val
25. end-if-else
26. repeat 13 to 25 while col < numCols
27. row++
28. outputFile ← length
29. repeat 6 to 28 while row < numRows

```cpp
#include <fstream>
#include <iostream>

class Rle{
    public:
    int numRow, numCol, minVal, maxVal;
    int startCol, startRow, length, current;

    Rle();

    void run(std::ifstream& i, std::ofstream& o);


};


int main(int argv, char** args){


    std::string input_filename = args[1];
    std::string output_filename = args[2];
    std::ifstream inp(input_filename);
    std::ofstream outp(output_filename);

    Rle rle;
    rle.run(inp, outp);

    inp.close();
    outp.close();

}

Rle::Rle(){
    startCol = 0;
    startRow = 0;
    length = 0;
    current = -1;
}

void Rle::run(std::ifstream& i, std::ofstream& o){
    i >> numRow;
    i >> numCol;
    i >> minVal;
    i >> maxVal;
    o << numRow;
```

```
    o << " ";
    o << numCol;
    o << " ";
    o << minVal;
    o << " ";
    o << maxVal;
    o << "\n";

    int next;

    for(int row = 0; row < numRow; ++row){
        i >> current;
        o << row;
        o << " ";
        o << 0;
        o << " ";
        o << current;
        o << " ";
        length = 1;
        for(int col = 1; col < numCol; ++col){
            i >> next;
            if(next != current){
                current = next;
                o << length;
                o << " ";
                o << row;
                o << " ";
                o << col;
                o << " ";
                o << current;
                o << " ";
                length = 1;
            }
            else{
                ++length;
            }
        }
        o << length;
        o << " ";
    }
}
```

OUTPUT 1:

4 12 0 9

0 0 0 6 0 6 7 6 1 0 0 2 1 2 4 5 1 7 0 5 2 0 0 5 2 5 3 2 2 7 6 4 2 11 0 1 3 0 3 11 3 11 7 1

OUTPUT 2:

12 12 0 9

0 0 0 6 0 6 7 6 1 0 0 2 1 2 4 5 1 7 0 5 2 0 0 5 2 5 3 2 2 7 6 4 2 11 0 1 3 0 3 11 3 11 7 1 4 0 0 12 5 0 0 2 5
2 1 8 5 10 0 2 6 0 9 5 6 5 0 3 6 8 9 4 7 0 8 7 7 7 5 5 8 0 0 12 9 0 0 12 10 0 2 6 10 6 1 6 11 0 0 12