<u>Program 2-1:</u> 2-dimensional Array (Task 1: Convert int 2-d array to char 2-d array)

Problem: Photos taken in space by the Pathfinder spacecraft are sent back to earth as a stream of numbers. Each number represents a level of brightness. A large number represents a high brightness level, and a small number represents a low level. Your job is to take a matrix (two-dimensional array) of the numbers and print it as a negative picture. If the numbers received represent levels of brightness, then one approach to generating a picture is to print a dark character (like a $) when the bright level is low, print a medium character (like a *) when the level is medium, and print a light character (like a blank) when the level is high.

- ‘ $ ’ : when bright level is greater than or equal to 5.
- ‘ * ’ : bright level is between 2 and 4 (including 2 and 4).
- ‘   ‘ : bright level is less than equal to 1.

Input: A text file “imagedata.txt" that contains 20 X 20 integer array of brightness.

Output: The program should display the following output.

```
Task 1

0 1 5 0 0 0 0 0 1 1 1 1 0 1 0 4 4 1 1 1
0 5 4 4 0 0 0 0 0 0 0 6 0 0 6 3 4 0 6 0
6 5 1 4 4 0 0 5 6 6 6 0 0 6 4 3 4 5 1 1
6 0 0 1 4 6 5 0 1 0 1 7 7 4 0 0 0 5 0 1
0 0 3 0 1 5 1 0 7 0 1 0 6 4 0 0 0 4 0 1
1 0 0 0 5 3 0 0 1 5 0 0 0 6 0 0 0 0 0 1
0 4 0 7 0 1 5 5 1 0 7 6 1 0 7 0 0 1 0 1
0 0 0 8 0 1 6 6 1 0 7 5 1 0 8 0 0 0 0 1
0 4 0 7 1 7 1 0 1 0 4 0 1 0 8 0 1 1 0 0
1 0 0 6 1 1 0 0 0 9 0 1 1 0 8 0 1 1 6 0
0 3 0 6 0 1 3 9 5 5 4 4 1 0 7 0 1 1 0 0
1 0 1 6 1 1 0 3 3 3 3 0 1 0 7 4 1 3 0 0
0 1 1 5 1 0 0 0 0 9 0 9 0 0 7 0 1 0 0 0
1 0 1 3 3 4 3 4 3 3 3 4 4 3 3 0 1 0 4 0
0 5 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
6 0 0 7 0 7 7 8 1 5 0 1 1 7 0 0 0 8 6 7
6 0 0 6 0 7 0 0 1 8 0 1 1 7 0 0 0 8 1 8
5 6 7 6 0 6 6 8 4 7 0 1 1 7 0 5 0 4 0 8
6 0 1 7 0 7 0 0 0 7 0 4 0 8 0 0 0 7 0 7
5 1 0 6 0 7 6 8 0 7 7 6 0 8 8 7 0 7 6 7

    $                         * *
  $ * *               $     $ * *   $
$ $   * *     $ $ $ $     $ * * * $
$       * $ $         $ $ *       $
    *     $     $       $ *       *
        $ *       $       $
  *   $     $ $     $ $     $
      $     $ $     $ $     $
  *   $   $         *       $
      $           $         $       $
  *   $     * $ $ $ * *     $
      $       * * * *       $ *   *
      $           $   $     $
      * * * * * * * * * * * *       *
  $
$     $   $ $ $   $       $       $ $ $
$     $   $       $       $       $   $
$ $ $ $   $ $ $ * $       $   $   *   $
$     $   $       $   *   $       $   $
$     $   $ $ $   $ $ $   $ $ $   $ $ $
```

Program 2-2: **2-dimensional Array (Task 2: Check/Correct error pixels)**

Problem: Unfortunately, errors in transmission sometimes occur. Thus your program should find and correct these errors. Assume a value is in error if it differs by more than one from each of its four neighboring values. Correct the erroneous value by giving it the average of its neighboring values, rounding it to the nearest integer.

Find and Correct Error Pixels

```
  5
4 2 5
  5
```

The 2 would be regarded as an error and would be given a correct value of 5 because the difference from its four neighboring values are 2 or more.

```
  5
4 3 5
  5
```

The 3 would be regarded as a correct value because the difference from the left neighboring value is 1.

You should assume that all neighboring values are correct when you check if it is in error or not. Also you should assume that all neighboring values are correct when you correct the error.

Neighboring Values

Current pixel: image[row][col]

- Up pixel: image[row-1][col]
- Down pixel: image[row+1][col]
- Left pixel: image[row][col-1]
- Right pixel: image[row][col+1]

Note that values on the corners or boundaries of the matrix have to process differently from the values on the interior.

Check Array Bounds Before Accessing Neighbors

```
-   if(row-1 >= 0)
        up = image[row-1][col];
-   if(row+1 < 20)
        down = image[row+1][col];
-   if(col-1 >= 0)
        left = image[row][col-1];
-   if(col+1 < 20)
        right = image[row][col+1];
```

Output: The program should print a negative image of the corrected picture.

Submission

You should submit the compressed file (.zip) including your source files (.cpp) to the Blackboard. No late submission will be accepted.