

Firstly, this program starts with taking input from user to modify existing terrain by using various methods. One of these methods takes input, such as “d3” and checks if the input is valid or not. Then, this method invokes another method to convert “d” and “3” to indexes. Then, by using these indexes another method modifies the terrain according to these indexes. Lastly, it draws the modified terrain to the console.

After taking all inputs, flood operation starts. In one of the methods, in a for loop, program invokes another method that does flood operation recursively. This recursive function firstly goes to neighboring blocks in a for loop to check whether they are lower than current block or not. If one of the neighboring blocks is lower than current block or equal to current block, it goes into them. When there is no lower neighboring block, current block will be flooded. While going back to previous block, if previous block has the same height with the flooded block, this block will be flooded too. Looping once, the lowest block will be flooded. One for loop invoking recursive function for every block, floods the blocks that has the lowest blocks in their lakes. Then another loop gives opportunity to the one higher block to be flooded. For example, if there are blocks that have the heights 1 and 2 in a lake, firstly, 1s will be flooded, then 2s will be flooded. Main for loop invokes this recursive function until there is no change in the terrain. This means every block that must be flooded is flooded.

In labeling phase, the algorithm is like the flooding algorithm. Program invokes recursive labeling function in a for loop which lays in while loop. Firstly, it starts from specific block and then checks if it is flooded or not. If it is flooded, it checks if its neighboring blocks are flooded or not. While doing this recursive function, it also adds these blocks to a ‘lake’ list, sets a label number for this block, and marks this block as labeled. At the end of the lake, main function adds the lake to ‘lakes’ list. Invoking this recursive labeling function lasts until all flooded lakes is labeled. After assigning each lake’s blocks separate label numbers, another function uses these numbers to convert them into letters. By doing this, labeling operation will be over.

In computing score phase, program firstly finds the block that has the minimum height among the blocks that surround the lake. Another method uses this number to compute the volume. Volume of the one block is equals the minimum surrounding height minus the height of the block. Then, by finding the volume of each lake, total score is computed.

In the last part of the code, program finally prints the final appearance of the terrain. In other words, it draws the console the flooded terrain with a different label on each lake. After that, it prints the score.