Main method:
Code starts with filling 2d array named matrix with one column more and one line more than the given terr ain.
It includes index values of every line which preceds the values that are in every line and has the last extra bottom line which has integer values of
corresponding chars. Those values will be casted to chars in the print method.
Modification inputs are taken in a for loo.If statements inside it check the modification according to its leng th and if it is in aa3 or a33 format. Any of these conditions not met,
it prints out invalid step prompt and decreases i 1 point to stabilize number of valid input taken to 10.If the modification is valid and falls under one of the if statements it first does the modification and
print the modified matrix.
Index numbers that are in the beginning of every matrix array line and the last line that contains integer va lues of each char correspond to a row are not useful for lake examination
therefore willBeSearched array only includes M*N size modified stone number array.
Finding which coordinates are in a lake;
2 big for loops investigate each point(only investigating nonbound points)A while loop is opened for every point and does not break unless lake flow reached a boundry point or lake stack is empty.
For every point, including that point and every neighbor point that fits to the rules are pushed to the stack.
Neighbor points are found with a for loop that iterates 8 times and uses every value in
directions array.It helps to get the coordinates of neigbor points by simply adding and extracting. If the nei ghbor's value is small or equal to the center's value and never been added to the lake stack
in that specific center points investigation process, neighbor mets conditions.After pushing to the stack an d marking added elements as true process of all neighbor points end. All of the stack will be checked whe ther
a boundry point has met the conditions and pushed to the stack.If so, while loop is breaked and the point we were currently looking for is not in a lake. If every point in the stack is
not a boundary one element of the stack is going to be popped and its coordinates will be used in neigbou r searching for loop(exactly like a liquid flowing).Added boolean array becomes useful exactly at this point .
Also pay attention that the value we use to compare neigbors is the center coordinates value which does not change when lSearcing and rSearching are updated.
If there is no more point to pop from the lake stack- which means at the last point we investigated that for ms the lake not push anything new(its coordinates were came from the preceding lake coordinate but that was marked
as added therefore neigbor investigating for loop (with the directions array) not pushes it even if its stone value mets the condition).
Unless a boundry point is not pushed to the stack throughout investigations, at a point stack becomes em pty which means a lake was formed,that point in the boolean array is marked and it is a part of a lake.
Finding separate lakes' algorithm is similar to finding coordinates that are in a lake but this time checked b oolean is not created for each center point and also center points(investigated points) are only the ones th at
are in a lake (booleanArray value is true).
Placing checked boolean outside 2 for loop helps us not to find each lake multiple times
Stack helps us to tell when finding coordinates of each lake stops because the fact that it becomes empty but those needed values are hold within a lakeList. That lakeList's modification will be finished after stack becomes empty
(means no more points left to add) and at that instant it will be added to the lakeLists array.
Final score calculation works with holding the sum of the stones of each lake in totals array, holding how many points create each lake in numbers array and creating a valueList every lake.
ValueList has every stone value that surrounds each coordinate of a lake (lake points next to that coordin ates are not added)Some values are added to the valueList multiple times but that does not matter becau se min value of all
is going to be taken(which height the liquid in my lake can reach the most)
Each element in numbers list is multiplied with each corresponding valueList's minimum element. That is t he maximum volume a lake might hold-assuming there is no stone inside the lake.

Values of total stones in every lake is going to be extracted from the maximum volume for every lake. Printing of the final 2d arrayList has done seperately and only once. ıts process includes naming puddles as (A-ZZ), naming each coordinate’s rows with(a-zz) and printing lines with index numbers.