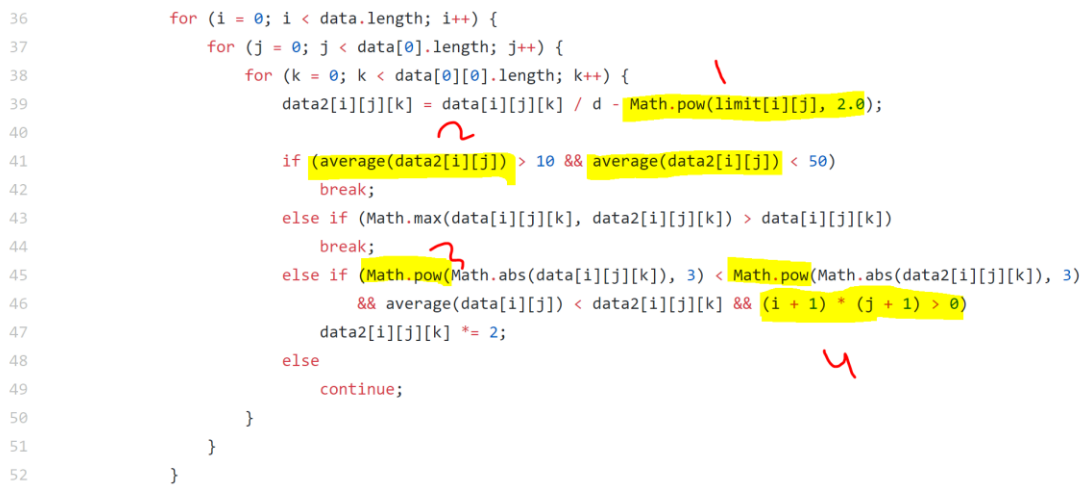
Part(1)



```
36  for (i = 0; i < data.length; i++) {
37      for (j = 0; j < data[0].length; j++) {
38          for (k = 0; k < data[0][0].length; k++) {
39              data2[i][j][k] = data[i][j][k] / d - Math.pow(limit[i][j], 2.0);
40
41              if (average(data2[i][j]) > 10 && average(data2[i][j]) < 50)
42                  break;
43              else if (Math.max(data[i][j][k], data2[i][j][k]) > data[i][j][k])
44                  break;
45              else if (Math.pow(Math.abs(data[i][j][k]), 3) < Math.pow(Math.abs(data2[i][j][k]), 3)
46                      && average(data[i][j]) < data2[i][j][k] && (i + 1) * (j + 1) > 0)
47                  data2[i][j][k] *= 2;
48              else
49                  continue;
50          }
51      }
52  }
```

As fig() show ,

Case (1) : Math.pow(limit[i][j], 2.0)

can be moved to line 38, that lead be calculated 2 instead of 3 times, with no changing in its algorithm .

double limit=Math.pow(limit[i][j], 2.0);

then use **“limit”** variable in line 39 as must be

Case(2) : average(data2[i][j]) > 10 && average(data2[i][j]) < 50

average can be calculated once time instead of double times, and move it to line 40.

averageData2=average(data2[i][j]);

then use **“averageData2”** variable in line 41 as must be.

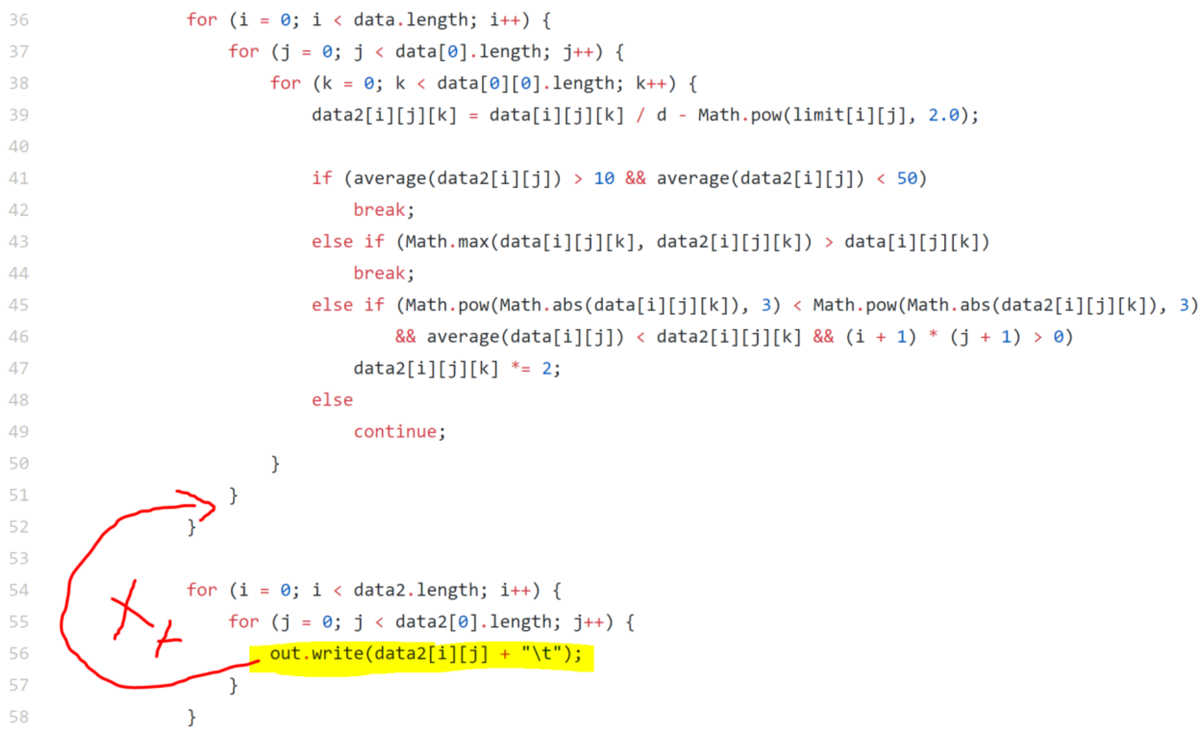
Case(3) : (Math.pow(Math.abs(data[i][j][k]), 3) < Math.pow(Math.abs(data2[i][j][k]), 3)

Math.pow can be deleted and check only with Math.abs values as here **Math.abs(data[i][j][k]) < Math.abs(data2[i][j][k])** , because one of math theories say that if x>y then $X^3 > y^3$**.**

Case(4) : (i + 1) * (j + 1) > 0

Can be removed , there is no need for using it, always will be greater than zero, that i & j start their loops from zero.

Part2



```
36  for (i = 0; i < data.length; i++) {
37      for (j = 0; j < data[0].length; j++) {
38          for (k = 0; k < data[0][0].length; k++) {
39              data2[i][j][k] = data[i][j][k] / d - Math.pow(limit[i][j], 2.0);
40  
41              if (average(data2[i][j]) > 10 && average(data2[i][j]) < 50)
42                  break;
43              else if (Math.max(data[i][j][k], data2[i][j][k]) > data[i][j][k])
44                  break;
45              else if (Math.pow(Math.abs(data[i][j][k]), 3) < Math.pow(Math.abs(data2[i][j][k]), 3)
46                      && average(data[i][j]) < data2[i][j][k] && (i + 1) * (j + 1) > 0)
47                  data2[i][j][k] *= 2;
48              else
49                  continue;
50          }
51      }
52  }
53  
54  for (i = 0; i < data2.length; i++) {
55      for (j = 0; j < data2[0].length; j++) {
56          out.write(data2[i][j] + "\t");
57      }
58  }
```

out.write(data2[i][j] + "\t"); can be moved to line 51, before second for loop ends as show in fig(), data2 array has the same length of data, so that can be printed directly after ending calculating , data2[i][j] use 2 dimension so need be printed in second loop, so, will be in line 51.