

**Lab3****Deadline: In lab on Oct 5****Requirements**

lab3.c skeleton file is provided for you. Write a function that rotates and zooms a 2D array of integers. The signature of the function is:

```
void rotateArray(char **arr, int n, int rows, int cols);
```

```
void zoomArray(char **arr, int n, int *rows, int *cols);
```

You need to implement a function that rotates this array by an angle specified by the command line argument. The input array size is limited to the maximum of 30x30. You could get an input that is smaller than 30x30. You will always get an input size whose result array will fit in 30x30.

`int n` in `rotateArray` function is the angle in degrees that the array will be rotated. This value can be any positive or negative value that is multiple of 90 degrees. `int n` in `zoomArray` function is the expanding/shrinking factor. Below will give you an example with 2 as `n` value in `zoomArray`.

Here is a 2x2 example of the input:

```
11
22
```

Here is the output of running the input file through your program when it is rotated 0 degrees and a zoom factor of 2. The original array orientation is retained, yet the whole image is enlarged by a factor of 2. Make sure that enlarging the image originates from the center of the image as in the example below.

```
1111
1111
2222
2222
```

Let's assume another example of 4x4 example with a zoom factor of 0.5. Here is the input:

```
1111
1111
4444
4444
```

The output will be:

```
11
44
```

You might wonder if an input such as below will be given with a zoom factor of 0.5:

```
1111
2222
3333
4444
```

The answer is no since there is no way to make this shrink to 2x2. Since you will be guaranteed such conditions, this should make the algorithm simpler.

You will be guaranteed to have an input that is **in squares**. Any character will be used in the array in this lab. Also, the zoom factor will be guaranteed to be a number that can make the image shrink or expand.

Lastly, note that the function parameters are different in this lab. We are using array pointers instead of using array with []. Please review the different and understand what the differences are. This is main reason why the output print function has moved to the main function unlike the previous lab.

### How to Compile and Run

```
gcc lab3.c -o < output executable>
./<executable> <input file> <angle> <zoom factor>
```

Now you will be able to see the output on your console.

### Restrictions

- You are not allowed to write any additional `printf` statement anywhere in the file.
- You are not allowed to modify any part of the code except `rotateArray` function, `zoomArray` function, and `a_num` variable.
- If you have any doubt, ask me during the lab session.

### Grading

Any grading failure due to not following instructions will result in 0.

- (1 point) All files are submitted correctly using the instructions below.
- (3 point) Generate a correct solution to the problem(s) in this lab. Three test inputs will be used.

### Submission Files

- You must push only one .c file named: **lab3.c** (case sensitive).
- Make sure to update your A number. Look at the top of the file and write your A number including leading 0's.
- Github link is in learning hub.