Online on Pointers

Section: A1+A2 Time: 50 minutes

Problem 1: Rotate a Matrix by 90 Degrees Clockwise

You are given a 2D square matrix NxN of integers. Write a function to rotate the matrix by **90 degrees clockwise** using **only pointer arithmetic**. You must not use array indexing (e.g., matrix[i][j] or matrix[i*N + j]). All memory accesses must be through pointers.

Constraints:

- `2 <= N <= 10`
- Use dynamic memory allocation to initialize the matrix.
- You can only use pointer dereferencing and pointer arithmetic.

Example:

Input:
3
1 2 3
4 5 6
7 8 9

Output:
7 4 1
8 5 2
9 6 3

See Problem 2 on the following page.

Problem 2: Decompress String

Write a function that takes a character array (`char *encoded`) containing a compressed string, and populates a new string (`char *output`) with the decompressed result.

The compression format follows the rule:

- A letter followed by a number indicates that the letter should be repeated that many times.

Use only **pointer arithmetic** to traverse and process the strings. Do **not** use array indexing (`encoded[i]`, `output[j]`, etc.).

Constraints:

- The numbers can be any positive value. If it is 0, remove the corresponding character. First try to handle numbers between 0-9 and later think how you can adjust that for multi-digit numbers.
- Use dynamic memory allocation to allocate memory for the output buffer.
- Only use pointer dereferencing and pointer arithmetic.

Example:

Input: a3b2c1
Output: aaabbc

Input: a10b0c5
Output: aaaaaaaaaaccccc

Note:
In C, a string is just a `char` array ending with a special escaped character: `'\0'` (null terminator).

This character `'\0'` marks the end of the string. Without it, functions like `printf`, `strlen`, etc. will keep reading memory and may print garbage or crash.

Always:

- Allocate an extra byte for the null terminator (size of array +1 in `malloc`)
- Set it manually at the end of your string

Example: Suppose, the length of the string is 7

```
char* str = (char*)malloc(8) // Allocate one extra byte
// Do whatever you want with str
// Add this ‘\0’ character at the end.
*(str+7) = ‘\0’
```