DSA Lab 4 - Pointers

CSE102

February 7, 2025

Instructions

- All code must be written in C only.
- There will be no partial marks unless specified.
- Submit only the `.c` files. Do not submit `.exe` or binary files.
- Use appropriate pointer notation instead of array indexing wherever required.
- Ensure proper memory management (no memory leaks).

Problem 1: Matrix Manipulation (5 Marks)

Instructions:

- The program should take two integers, `rows` and `cols`, as input to define the matrix size.
- Initialize the matrix such that

$$\texttt{matrix}[i][j] = (i+1) \times (j+1).$$

- If `rows == cols`, transpose it in-place using pointer swapping.
- If `rows` $\neq$ `cols`, allocate a new matrix and compute the transpose.
- Compute the sum of all elements in the matrix.
- Free dynamically allocated memory.

Example Input:

```
Enter number of rows: 3
Enter number of columns: 4
```

Example Output:

```
Original Matrix:
1  2  3  4
2  4  6  8
3  6  9 12

Transposed Matrix:
1  2  3
2  4  6
3  6  9
4  8 12

Sum of all elements: 60
```

Problem 2: Shake the Array (5 Marks)

Instructions:

- Given a string of size N, perform in-place left and right shift operations.
- Perform the left shift operation K times and then the right shift operation L times.
- The array should be modified in-place (O(1) space) and the operation should run in O(N) time.
- Only pointer-based string manipulation is allowed. Use only `stdio.h`.
- Note: K and L can be greater than N.

Example Input:

```
HelloWorld
3 5
```

Example Output:

```
loWorldHel
ldHelloWor
```

Example Input:

```
imaginecrashes
20 14
```

Example Output:

```
ecrashesimagin
ecrashesimagin
```