Lab Assignment -2

Name: Shubhang Tripathi
Enrollment No: 18114074
Batch: O3

Problem Statement -1:

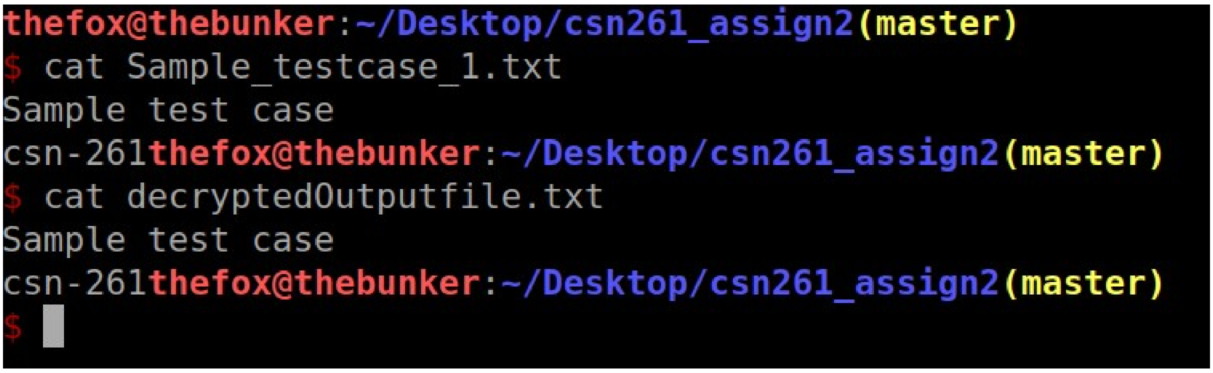
Write a program transpose.c that takes n, a, b, inputfile.txt in argv[1], argv[2], argv[3], and argv[4], respectively, applies the above encryption; and writes the result to outputfile.txt. Further, write a program inverseTranspose.c that decrypt the outputfile.txt and result in a new file named decryptedOutputfile.txt. Finally, write a program compareFiles.c to find the equivalence between the inputfile.txt and decryptedOutputfile.txt files. You may assume that n, a, and b are all small enough to fit into variables of type int. Your program should exit with a nonzero exit code if n is not at least 1 or if it is not given exactly four arguments, but you do not need to do anything to test for badly-formatted arguments. You should not make any other assumptions about the values of n, a, or b; for example, either of a or b could be zero or negative.

DATA STRUCTURES USED:

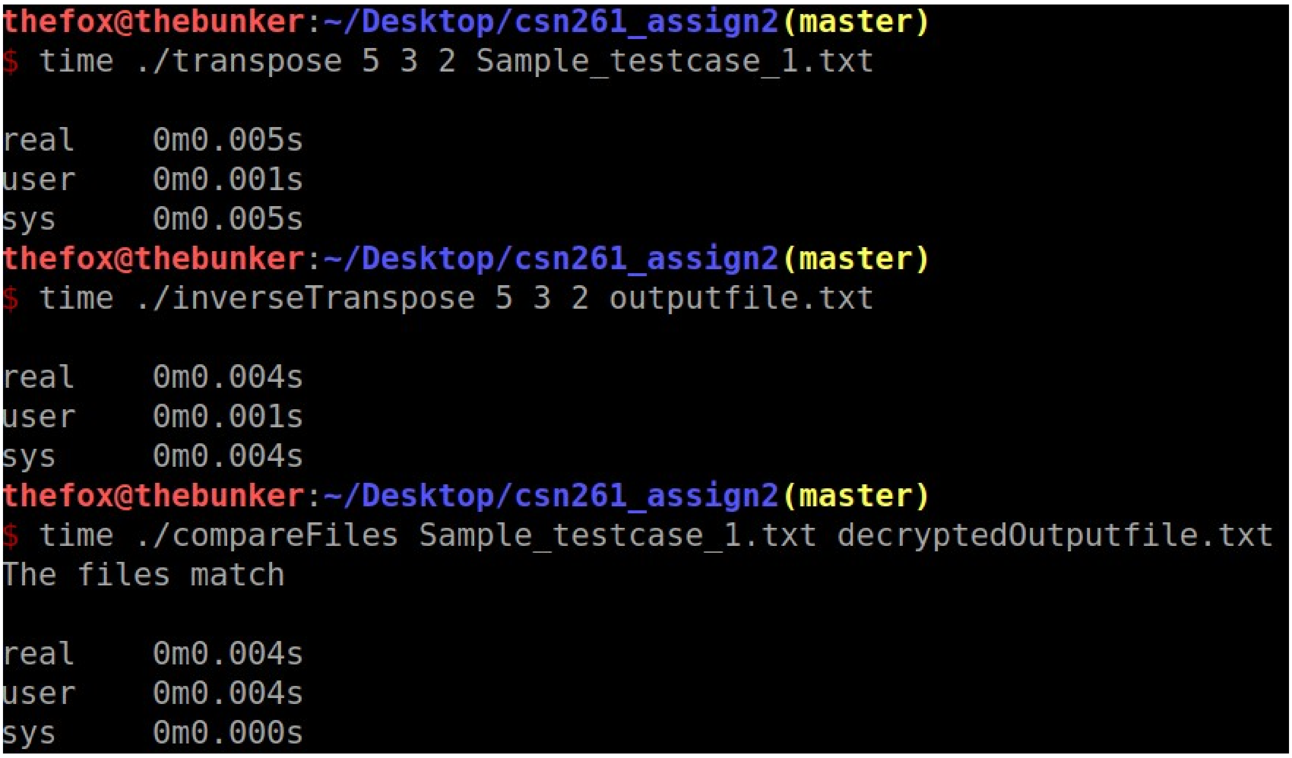
- Resizeable String
- One-Dimentional array

ALGORITHMS USED:

- Realloc used to dynamically increase capacity of string
- Transposition ciper implemented



```
thefox@thebunker:~/Desktop/csn261_assign2(master)
$ cat Sample_testcase_1.txt
Sample test case
csn-261thefox@thebunker:~/Desktop/csn261_assign2(master)
$ cat decryptedOutputfile.txt
Sample test case
csn-261thefox@thebunker:~/Desktop/csn261_assign2(master)
$ 
```



```
thefox@thebunker:~/Desktop/csn261_assign2(master)
$ time ./transpose 5 3 2 Sample_testcase_1.txt

real	0m0.005s
user	0m0.001s
sys	0m0.005s
thefox@thebunker:~/Desktop/csn261_assign2(master)
$ time ./inverseTranspose 5 3 2 outputfile.txt

real	0m0.004s
user	0m0.001s
sys	0m0.004s
thefox@thebunker:~/Desktop/csn261_assign2(master)
$ time ./compareFiles Sample_testcase_1.txt decryptedOutputfile.txt
The files match

real	0m0.004s
user	0m0.004s
sys	0m0.000s
```

Problem Statement -2:

Write a C program, MAT.c to represent any region (in image array representation), into its quadtree form.

- Print the Maximal square array where it should be filled in the following order: top-right, top-left, bottom-right and bottom-left quadrant, this should be done recursively for all the sub-quadrants. All the cells within a maximal square block should be filled with its corresponding block number.

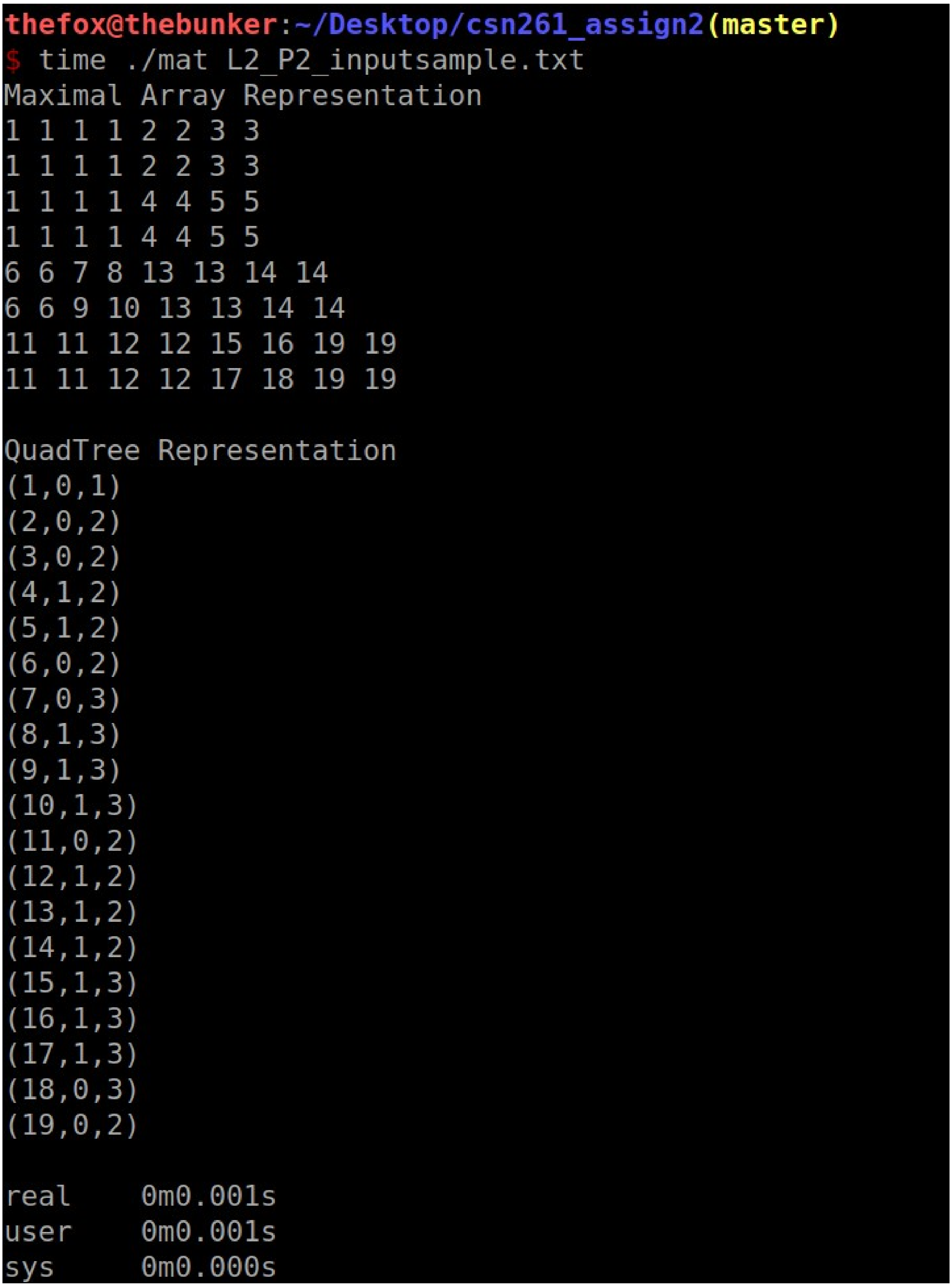
- Print the quadtree in the following manner, labels of leaf nodes, corresponding bit value and their level information (assuming the level of the root node to be 0), while traversing the quadtree in postorder.

DATA STRUCTURES USED:

- 2-Dimentional Dynamic Array
- Quadtree

ALGORITHMS USED:

- Mostly linear searching and O(n^2) functions have been used.



```
thefox@thebunker:~/Desktop/csn261_assign2(master)
$ time ./mat L2_P2_inputsample.txt
Maximal Array Representation
1 1 1 1 2 2 3 3
1 1 1 1 2 2 3 3
1 1 1 1 4 4 5 5
1 1 1 1 4 4 5 5
6 6 7 8 13 13 14 14
6 6 9 10 13 13 14 14
11 11 12 12 15 16 19 19
11 11 12 12 17 18 19 19

QuadTree Representation
(1,0,1)
(2,0,2)
(3,0,2)
(4,1,2)
(5,1,2)
(6,0,2)
(7,0,3)
(8,1,3)
(9,1,3)
(10,1,3)
(11,0,2)
(12,1,2)
(13,1,2)
(14,1,2)
(15,1,3)
(16,1,3)
(17,1,3)
(18,0,3)
(19,0,2)

real	0m0.001s
user	0m0.001s
sys	0m0.000s
```