Lab Test Week 9 : 2D 'Sorting'

Examination Rules

This is an exam. Please:

- Wait to be seated by a Lab. supervisor.
- Do not talk to anyone else.
- Do not use a Web browser, unless instructed to do so (e.g. to submit files to SAFE).
- Do not use an email connection.
- Do not look in other peoples directories.
- Do not use code written in 'groups' with others students.
- **Do** use text books, lecture notes, man pages etc.

If you do not obey the above rules you will be asked to leave.
People arriving late may not be allowed into the Lab.
You have one hour. There is additional time in case a server crashes or some other problem occurs.

Submitting Your Work

As you complete each part, submit it online using the SAFE submission system.
If you complete Part 1, submit a file called `partone.c`.
If you complete Part 2, submit a file called `parttwo.c`.

Let a lab supervisor know how many parts you believe you've completed.
No partial marks are available for each part. The only possible scores for this exam are 0%, 60%, or 100%.

Code Style

- Your code must compile without warnings using the gcc flags:

  ```
  -pedantic -Wall -Wextra -Wfloat-equal -ansi -O2
  ```

- Do **NOT** use any global variables.
- If you are using `rand()` in your code, feel free to set the seed of the random number generator to give different outputs each time your program is run, for instance using :

```
srand(time(NULL));
```

remembering to `#include <time.h>`

Part 1 (60%)

Write a **function** that fills up a square board, randomly, with an integer $0 \ldots 9$. Use a:
`#define N 20`
to define the size of the board. Write another function to print the board. The board may look something like:

```
36753562912709360626
18792023759228973612
93194784503610632061
55476569374525474430
78688431492068926649
50487172722610615949
09177115977673656394
81293908850963856115
98481030444476317596
21785741859753883189
64333860488897764303
09254059469224775481
28936802105110850646
25862847240629908131
10340391969338056640
04626756987282996027
61321599149107587048
04296104222055290283
80409196254499360502
94351743146942264128
```

Write a function to 'mutate' the board. Mutating is done like this:

- Choose two random locations which are **horizontally adjacent** (next to each other left-right).
- Swap these two numbers on the board if the left one is greater than the right one, numerically.
- Choose two random locations which are **vertically adjacent** (next to each other up-down).
- Swap these two numbers on the board if the upper one is greater than the lower one, numerically.
- Repeat the above steps (N*N*N) times.

Now print out the board. It should look something like :

```
00000000000001111224
00000001111111233456
00000111112222244456
00001122222333445666
00112222223333555667
01112223333334556678
01112223334445556779
01122333344445556789
01223334444455666789
01223344445556666789
01223344455666667789
01224444456666777889
01234455566677777889
01234555666677788899
01234555666778888899
12234566677788888999
12345567777888889999
12445677788888899999
34446678889999999999
46678899999999999999
```

Ensure that if you change the size of your array, by changing your `#define` that the program still operates correctly.

Part 2 (40%)

Adapt the code above, so that the algorithm is:

- Choose two numbers at random locations on the board.
- Check that of these two numbers, the one closest to the centre of the array is numerically less than the number furthest away from the centre. If not, swap them.
- Repeat the above steps (N*N*N*N) times.

Once again randomise the array initially, and ensure that after changing your `#define` the program still works correctly.
When $N = 21$, the array may look something like:

99999888777778889999
99998766656666788999
99887655444455678899
99866544333444566889
98765433322233456789
87654332211222334567 8
86543211111112334568
76543211000011234568
76542210000001234567
76432110000001223467
76432110000001123457
76432210000001223467
76542210000001224567
86543211000011234568
86543221111122334568
87654332212223345678
98765433322233456789
98866544333344567889
99887655444455678899
99987666656666789999
99999888777778899999