

## CMPUT 350 Lab 9 Exercise Problems

---

Rules:

- You can use all course material and man pages, but no other information such as web pages, books, or written notes. Using other information sources during the exercise constitutes cheating.
- Your programs must compile without warning using

```
g++ -Wall -Wextra -O -g -std=c++17 ...
```

In case there are compiler warnings or errors, marks will be deducted.

- Test your programs with different values. For now, the speed of your program is irrelevant. So don't spend time on optimization
- You must check for the appropriate preconditions/postconditions. Your program shouldn't crash or have undefined behaviour (**hint**: use asserts)!
- Your programs must be well structured and documented. Use ctrl-x t in Emacs to pretty-print it. Marks are assigned to functionality, program appearance, and comments.
- In case your program hangs, use ctrl-c to terminate it.
- Remember that you need to include the appropriate header files. To find out which ones you need for specific functions such as printf, use then man command.

Submit your solution files `ttt2.cpp` on eClass under "Lab 9 / Submission".

**Important:** Submit often (the system will only accept solutions submitted before 16:50)

---

1. [25 marks]

In this exercise you will complete a program that solves tic-tac-toe states using **negamax search** (see AI part4 page 27). I.e., your program will answer who wins in a given game situation assuming perfect play on both sides.

In file `ttt2.cpp`, which is provided, implement the parts marked with

```
// ... implement
```

and test your program for correctness and proper cleanup.

Use `./make2` to compile your program.

**IMPORTANT:** Ensure that your code works for N-by-N tic-tac-toe, i.e., changing value N to other values than 3 should work, e.g.

```
static constexpr int N = 4;
```

The winning condition for N-by-N ttt is having N pieces of the same colour in any row, column, or main diagonal. Given that N can vary, you'll need loops to detect winning conditions.