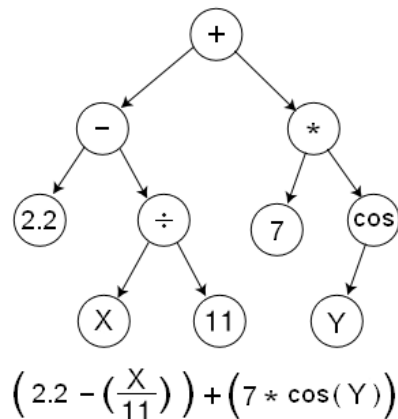


BIL443/553 Pattern Recognition
Homework 2

You are probably familiar with the **snake game**, where a “snake” moves in a rectangular grid (with no obstacles), eats up food (occupying a single cell), and grows longer every time it eats, avoiding crashing on itself or to the external walls of the grid. The goal is to eat as many pieces of randomly occurring food as possible before growing too long.

Your task is to write a program, that will write a snake playing program. This is genetic programming, a sub-field of genetic algorithms where every individual of the population is a simple program represented by a tree of commands as in the below example:



Programs represented by trees are made up of terminals (leaves) and functions (internal nodes). In this particular case your terminals will be **move forward**, **turn left**, **turn right**. You are free to choose/imagine your own functions (e.g. **if_food_ahead**, etc).

0. Read up online sources on genetic programming
1. Think of functions to use that could be useful for a snake’s survival.
2. Design a fitness function to evaluate every candidate program (e.g. how many pieces of food it has eaten? What if it didn’t manage to eat any food before it crashed?)
3. Design operators for mutation/crossover of candidate programs (i.e. population individuals)
4. Implement it all and report your findings. What kind of programs did your program write? How well did they perform? For how long did your snake survive while executing the program that your program wrote?

You are expected to submit a report, an executable (terminal based is ok, GUI is optional + 30 points) and your source code (C/C++, Java or Python).

Good luck.