**Main Program:**

Responsible for general program flow will constantly loop calling the Turtle Movement Module's moveTurtle function while it returns 0. And halts if moveTurtle returns a negative number. Must print an error message corresponding to each error code. Program will accept as an argument the schematic file to be processed. Will call moveTurtle(true) if the –excavate flag is set.

Notes:

* Return codes: 1 = Schematic complete, 0 = Nothing to report -1 = Area is not clear, -2 = Out of materials
* Possible flags: -excavate (Causes the turtle to excavate if it's path is blocked)

**Modules:**

* Schematic Reading Module
* Turtle Movement Module
* Item Placement Module

**Planned Future Modules:**

* Area Clearing Module
* Area Filling Module

**Schematic Reading Module:**

Responsible for reading a schematic file and reading the file into a 3 dimensional array. The turtle will be assumed to start in a specific corner with a specific facing. Each element of the array will be a number between 0 and 15 corresponding with the inventory slots in the turtle. Combinations of numbers should also be possible, such as 4,7,9 or 1-4 indicating that multiple slots contain the same material and any of those slots can be used. A cell with an X indicates that cell is supposed to be air

Must have a function called getNextMovement(int aX, int aY, int aZ) that can be called with current x,y,z coordinates of the turtle and return direction of the turtle's next movement

Must have a function called getItemToPlace(int aX, int aY, int aZ) and return the slot number/numbers (return an array or tuple regardless of number of elements) to be placed in that cell

Notes:

* The x,y,z coordinates will not be the x,y,z coordinates of the greater Minecraft world, but relative to the schematic
* The items will be placed directly below the turtle to ease navigation constraints, this should not affect code in this module, but keep it in mind

**Turtle Movement Module:**

Responsible for moving the turtle, detecting whether movement is possible, and excavating if the way is blocked and an argument is passed to the program.

Must have a function called moveTurtle(bool aExcavate = false) that will use the Shematic Module's getNextMovement function and

1. Check whether that block is cleared or open
2. Excavate that block if the 'excavate' argument is passed, otherwise return -1 (Area is not clear)
3. Move to the next block
4. Use the Item Placement Module's placeItem function to place whatever item is needed in the needed slot. Return -2 (Out of items) if placeItem returns -1
5. Return 1 (Schematic completed) if getNextMovement returns 1 or 0 if no errors occured

Notes:

* Make sure to loop through steps 1 and 2 until step 1 is satisfied, in case the turtle is excavating sand or gravel

**Item Placement Module:**

Responsible for placing selecting the correct item slot and placing a block in a specified element of the schematic

Must have a function called placeItem(int aX, int aY, int aZ) that will use the Schematics Module's getItemToPlace and the select the correct inventory slot and place the item below the turtle

Notes:

* When given multiple inventory slots, this function must cycle through the slots as the slots are used up