Design:

classes:

Ant

Board

Space

functions:

main()

menu()

runSimulation()

class Ant

integer m\_antRow

integer m\_antColumn

enum antOrientation

has 4 values: Left, Right, Up, Down

void rotate(char direction) rotates the ant left or right depending on the passed direction

int getRow()

int getCol()

void setRow(int row)

void setCol(int col)

class Space

constructor that initializes the spaceColor to white and the displayedSpace to ‘ ‘.

enum spaceColor

has 2 values: White and Black

character displayedSpace

getSpaceColor()

setSpaceColor()

getDisplayedSpace()

setDisplayedSpace()

class Board

constructor that creates the dynamic array of Space objects and places the ant.

2D array of Space objects representing the board: m\_board

Ant member object: m\_ant

updateSpace(int row, int col)

changes the character displayed in the space to reflect its current status. (white, black, or occupied)

void moveAnt()

if the space the ant is currently occupying is white turn it right 90° and change the space to black. if it is black, turn it left 90° and change the space to white.

change location of ‘ant’ based on orientation: up (row decreases by 1), down (row increases by 1), left (col decreases by 1), right (col increases by 1). If the ant is going to move off the board the row and column don’t change.