cis1.5-spring2007-sklar, lab II, part 1

instructions

- This is the first part of the lab/homework assignment for unit II.

- The entire assignment will be worth 9 points.

- The first part is worth 4 points and will be distributed and worked on in class on Wednesday February 28.

- The second part is worth 5 points and will be distributed and worked on in class on Wednesday March 7.

- **Both parts together are due on Monday March 12** and must be submitted by email (as below).

- **Follow these emailing instructions:**

  1. Create a mail message addressed to *sklar@sci.brooklyn.cuny.edu* with the subject line **cis1.5 hw2**.
  2. Attach ONLY the **.cpp** files for each part, as outlined below.
     DO NOT ATTACH THE **.cbp** (CodeBlocks Project) files!
  3. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (which can make it a lot harder for me to grade your work — grrrr!)

## 1   calculating roomba's distance to home.

(2 points)

- Look at the **roomba4.cpp** example which we discussed in class on Monday February 26.

- Create a new project in CodeBlocks and enter the **roomba4.cpp** code in that project.

- Compile, build and run it to make sure it works as you expect it to.

- Then modify the program so that the robot's *home* location (where its charging station is stored) is not at $(0,0)$ but at a *randomly* chosen location.

- *Hint:* get the program to set randomly the values of xhome and yhome, like we did in class on the blackboard.

- Use the abs() function that we discussed in class on Monday to account for negative numbers when computing the Manhattan distance.

- Make sure that the program calculates the Euclidean and the Manhattan distances back to home correctly!

- Make sure the program compiles, builds and runs as expected.

MORE ON THE NEXT PAGE.....

2   considering all the cases.

(2 points)

If the home location and roomba's initial location are chosen randomly (as above), then there are a number of possibilities for how these two locations are related to each other on the grid. Consider the drawings below:

| (a) | (b) | (c) | (d) |
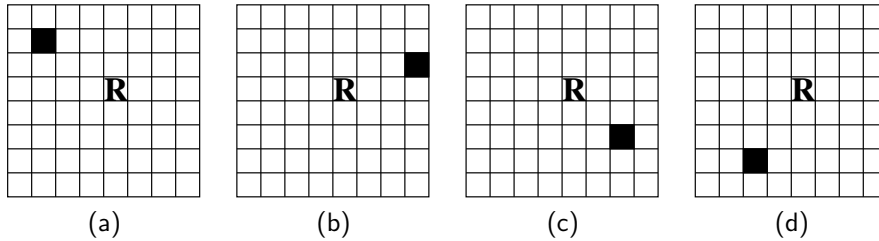
In these drawings, the robot is represented by the letter **R** in the grid and the charging station (home) is represented by a black square. In figure (a), the home is to the northwest of the robot. In figure (b), the home is to the northeast of the robot. And so on.

- Create a new project and copy your code from the previous step (above) into the new project.

- Modify the way your program calculates the Manhattan distance from the robot to the home, WITHOUT USING THE `abs()` FUNCTION.

- In order to do this, you need to consider all the possibilies!

- *Hint:* use `if-else` statements to make sure that when you subtract two numbers from each other, the result is always greater than or equal to 0.

- *Hint:* there are more possibilities than just the four drawings shown! What are they? Try drawing them on paper to help you figure out your program.

lab II, part 2

... will be distributed in class on Wednesday March 7.

Appendix A

```cpp
/**
   roomba4.cpp
   26-feb-2007/sklar

   this program expands on roomba3.cpp by calculating the distance the
   roomba needs to travel from its randomly selected initial location
   to its home location (0,0).
*/


//----- section 1: include C++ library definitions -----
#include <iostream>
#include <math.h>
#include <stdlib.h>
#include <time.h>
using namespace std;

//------ section 2: declare variables -----
int x; // robot's x position
int y; // robot's y position

//----- section 3: declare methods -----
int display() {
  cout << "the roomba is at location (" << x << "," << y << ")\n";
  return 0;
} // end of display()

//----- section 4: define main method -----
int main() {
  // declare variables
  double d;
  // initialize random seed
  srand( time( NULL ));
  // find random initial location
  x = rand() % 10;
  y = rand() % 10;
  display();
  // compute distance to "home" (0,0)
  // euclidean distance
  d = sqrt( (double)( x*x + y*y ));
  cout << "Euclidean distance to home is: " << d << endl;
  // manhattan distance
  d = x + y;
  cout << "Manhttan distance to home is: " << d << endl;
  //  double f = sqrt( 4.0 );
  double f = sqrt( (double)4 );
  cout << "f = " << f << endl;
} // end of main()
```