fahad hussain

CE315 Assignment 1

Contents

[Introduction 2](#_Toc95988427)

[Task 1 3](#_Toc95988428)

[Flowchart 5](#_Toc95988429)

[Task 2 6](#_Toc95988430)

[Flowchart 7](#_Toc95988431)

[Appendix 8](#_Toc95988432)

[Task 1 8](#_Toc95988433)

[Task 2 9](#_Toc95988434)

# Introduction

A mobile robot is a system/machine that usually contains sensors and other components to allow it to perform its intended task, There are usually two types of Mobile robots, Autonomous Robots are robots that use Sensors and Artificial intelligence pathing algorithm, This allows them traverse a given environment without the guidance of an external controller, For instance, A underwater mobile robotic fish, This looks and behaves like a normal fish but will have sensors and cameras attached to it for navigation. This machine can be extremely useful in the right hands, A marine biologist could use this robot to gather information in any underwater environment, it can also be used to assess how healthy a coral is without disrupting the eco system, if a human being were to swim and record data manually it may lead to inaccurate results, this is because researchers can influence the results, such as changing the natural environment. For example, by being physically present the researchers can cause the behavioural differences of live specimens in the area or damage to sensitive structures.

A common type of robot that is typically used is a manually/semi assisted robots, these usually need to be navigated by an external controller, this controller will usually be able to see the environment that the robot is in as they tend to have cameras attached. An example of this could be the Mars Rover, since human beings are more restricted in space it makes more sense to use an external source to perform the same function that the person could be doing. A rover is a much safer approach for researchers to collect data in space than it is for them to manually do it themselves, the combination of artificial intelligence and an improvement in robotics can aid in scientists/researchers explore space much more effectively.

Another type robot that is commonly used are flying robots, these can have various purposes, for example these drones can have excellent vision on a given environment and are relatively quick making them extremely useful in search and rescue cases, If a natural disaster like an earthquake were to occur it would make much more sense and be quicker to look for survivors using drones than for people to manually perform the same task, this logic can also be applied to terrains that are dangerous for humans to enter for example an icy environment or an environment that is hazardous to humans (Radiation).

A lot of robots that are developed in this field tend to be biologically influenced by living animals, Robots can range from one-legged to even 8 legs, usually any robot that has 6 or more legs will have its design influenced by the biology of an insect, these robots can be used as a means to collect accurate data on the behaviours of small hive minded creatures such as Bees, Ants and many more biological species, robotics is a field that aids several other industries.

The industry that has benefited most from robotics in the last decade would be the manufacturing industry, since the introduction of automated robotic machinery it makes much more sense financially and economically for businesses in the manufacturing industry to use machines instead of people, due to the fact that they are more efficient, cost effective and do not feel tiredness or hunger and are in the long term cheaper than labour, these are all attributes that make mobile robots so desirable to businesses, especially in the manufacturing industry.

# Task 1

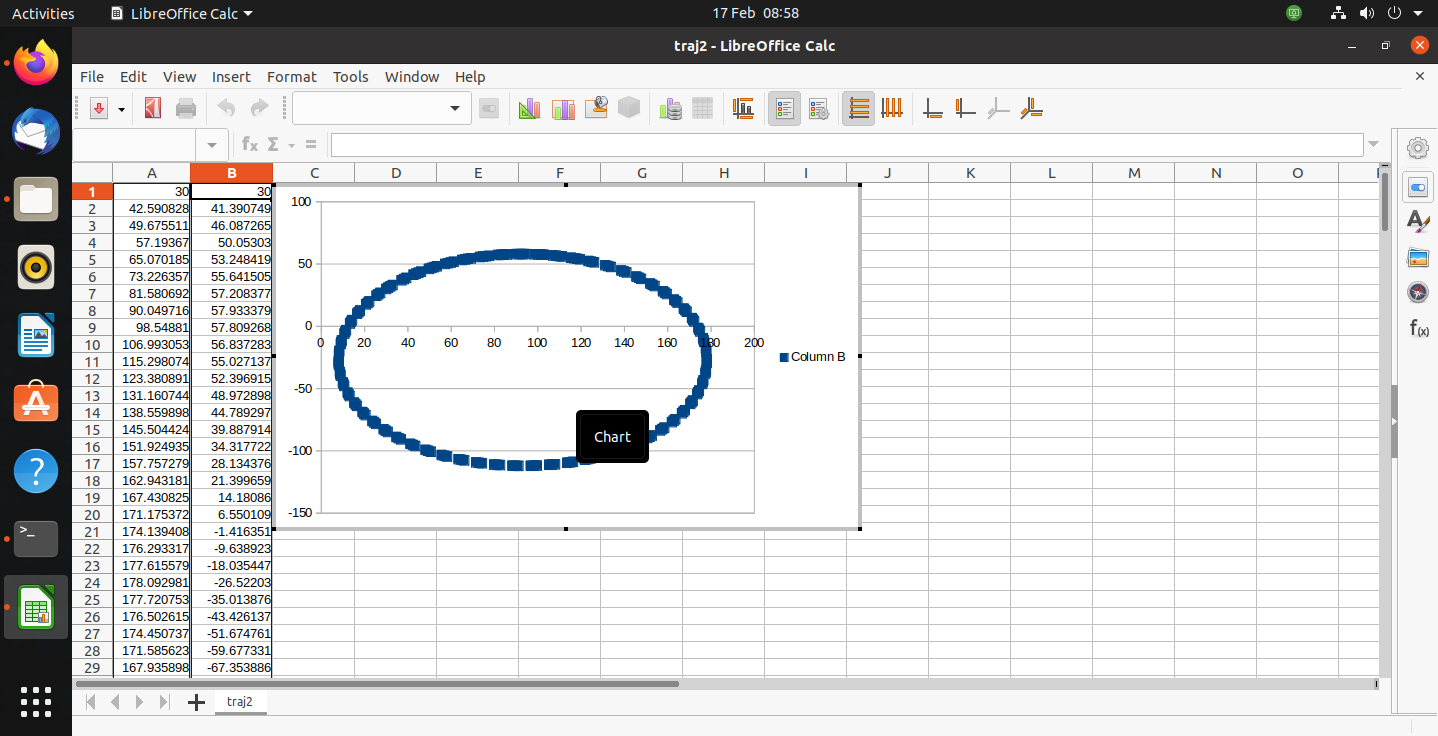


Figure 1

This was the graph that was created when the velocity of the left wheel was 10cm/s and the right wheel was 8cm/s.

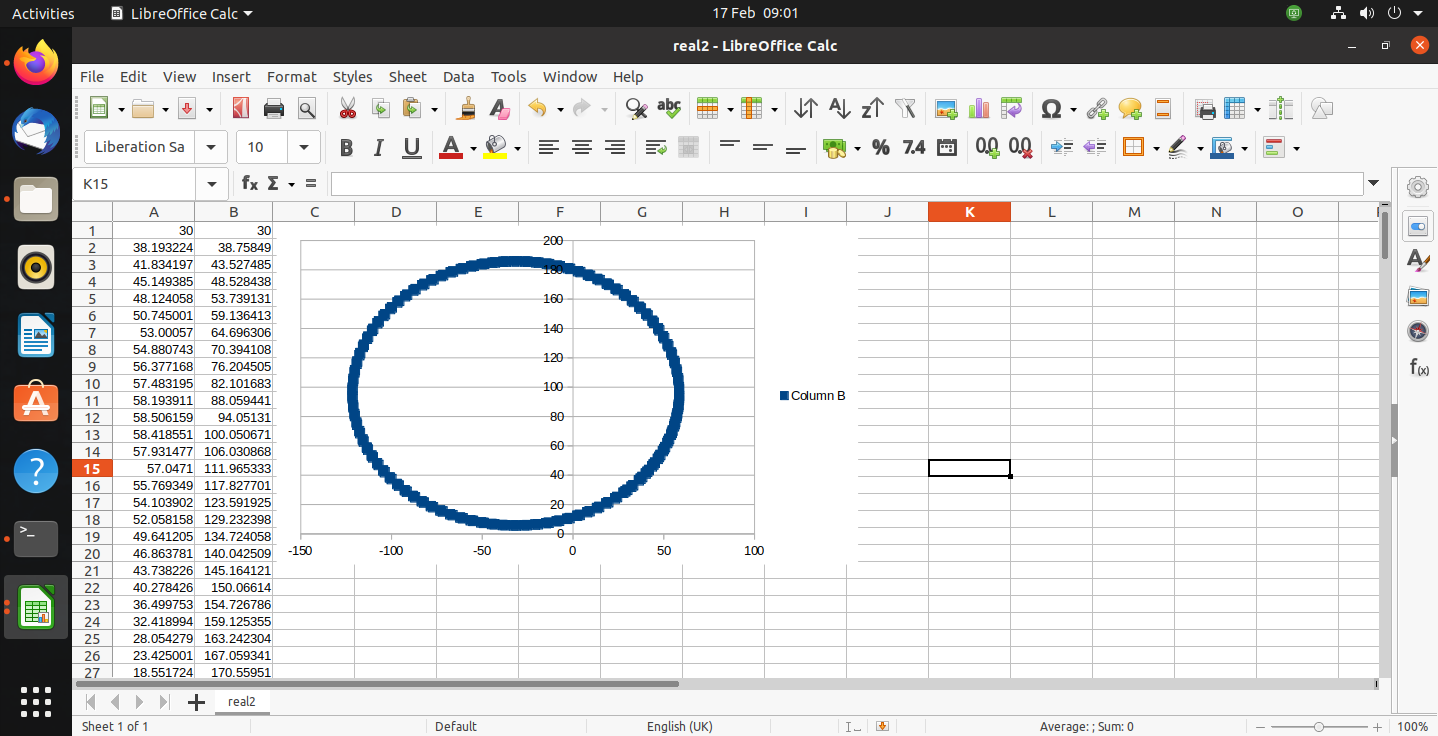


Figure 2

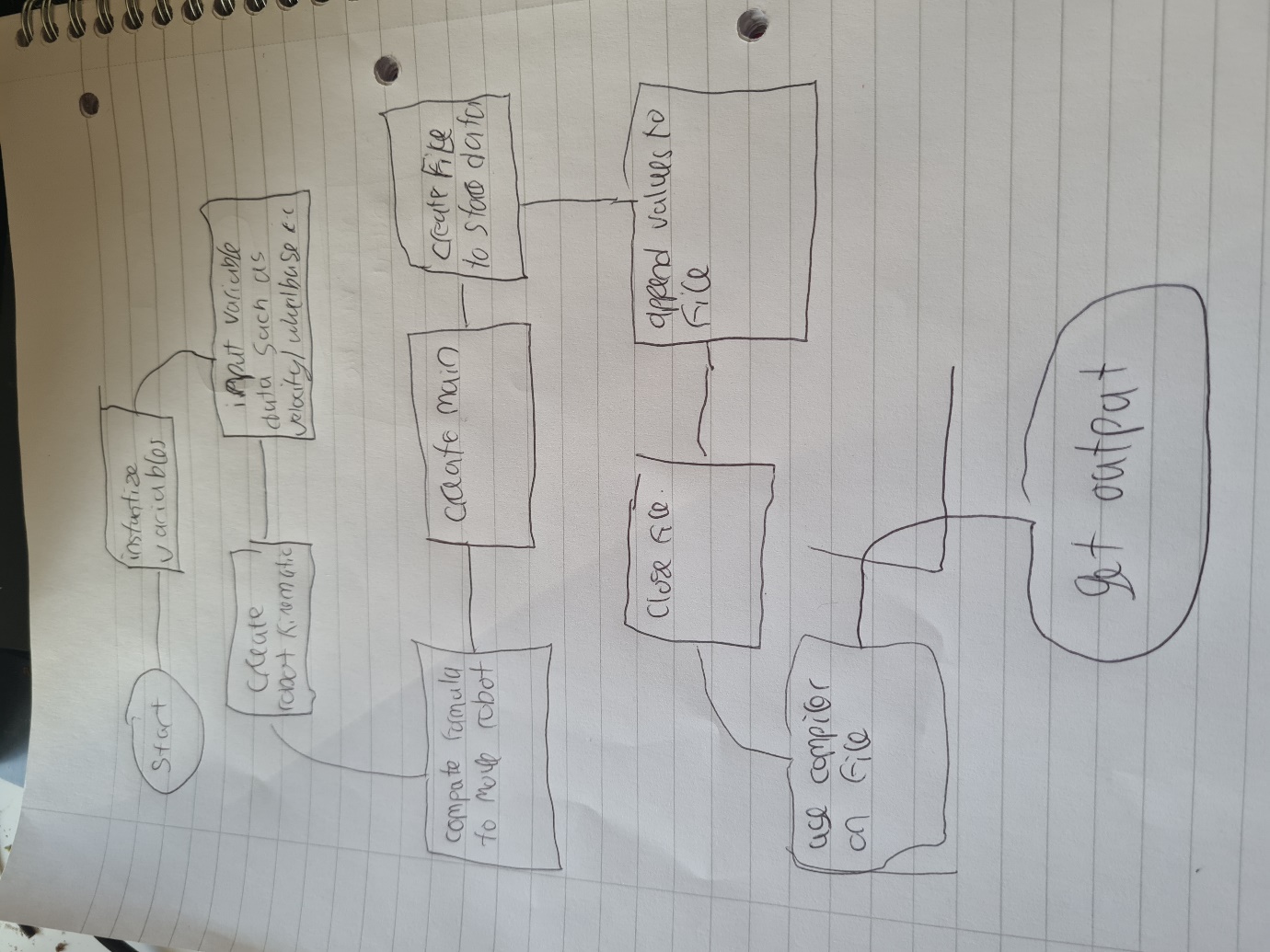
This was the graph that was created when the velocity of the left wheel was 5 cm/s and the right wheel was 7cm/s.

A differentially driven robot tends to have two wheels which can have different turning rates and they also usually have a castor wheel. If the velocity of both wheels are the same then robot will move in a straight line, however if one wheel is moving at a different rate the robot will then move in the direction that the slower wheel is In so for instance if the right wheel is going at 20cm/s and the left wheel is going at 14cm/s then the robots trajectory will be towards the left hand side. There are other aspects that must be known to the controller when using a differentially driven robot, one being how far apart are the wheels from one another, this can also be known as the wheelbase, this data has already been provided. Another aspect would be the radius of the wheelbase which has also been given.

Once all the values are known to the controller, it can then be inputted into the formula that is required to make a differentially driven robot move, to find the value of x’s position it is x(k+1)=x(k) + Vr+Vl/2 \*cos (theta)\* cycle time, To find the value of y’s position it would be y=(k+1)=y(k)+Vr+Vl/2\*sin(theta)\* cycle time. Theta also known as the direction that the robot will be traversing in can be calculated by Theta(k+1)= Theta(k) +Vr -Vl/W \*cycle time, once the kinematics calculation is implemented all that’s left is to write data onto a text file, the number of values can be specified to k which is already set to 200.

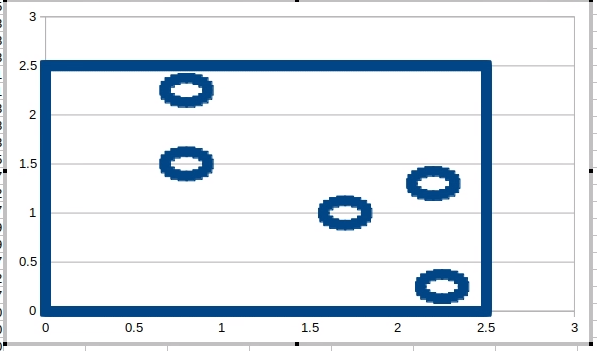
Looking at the figures above, in figure 1 the highest y value that the robot went to was 50 and the highest x value was 180, In figure 2 the largest y value was 180 and the highest x value was roughly 50.

The main difference between the two graphs were that the left wheel value of the first figure was almost double the second figures value, the difference between the velocity of the left wheel and the right wheel was the same in both tests even though they are different values. In figure 1 the robot will traverse more to the right wheels side as the right wheel is slower than the left wheel (10s/cm<8s/cm). in figure 2 the robot will move in the left wheels direction as the left wheel is slower than the right wheel (5s/cm<7s/cm).



## Flowchart of task 1

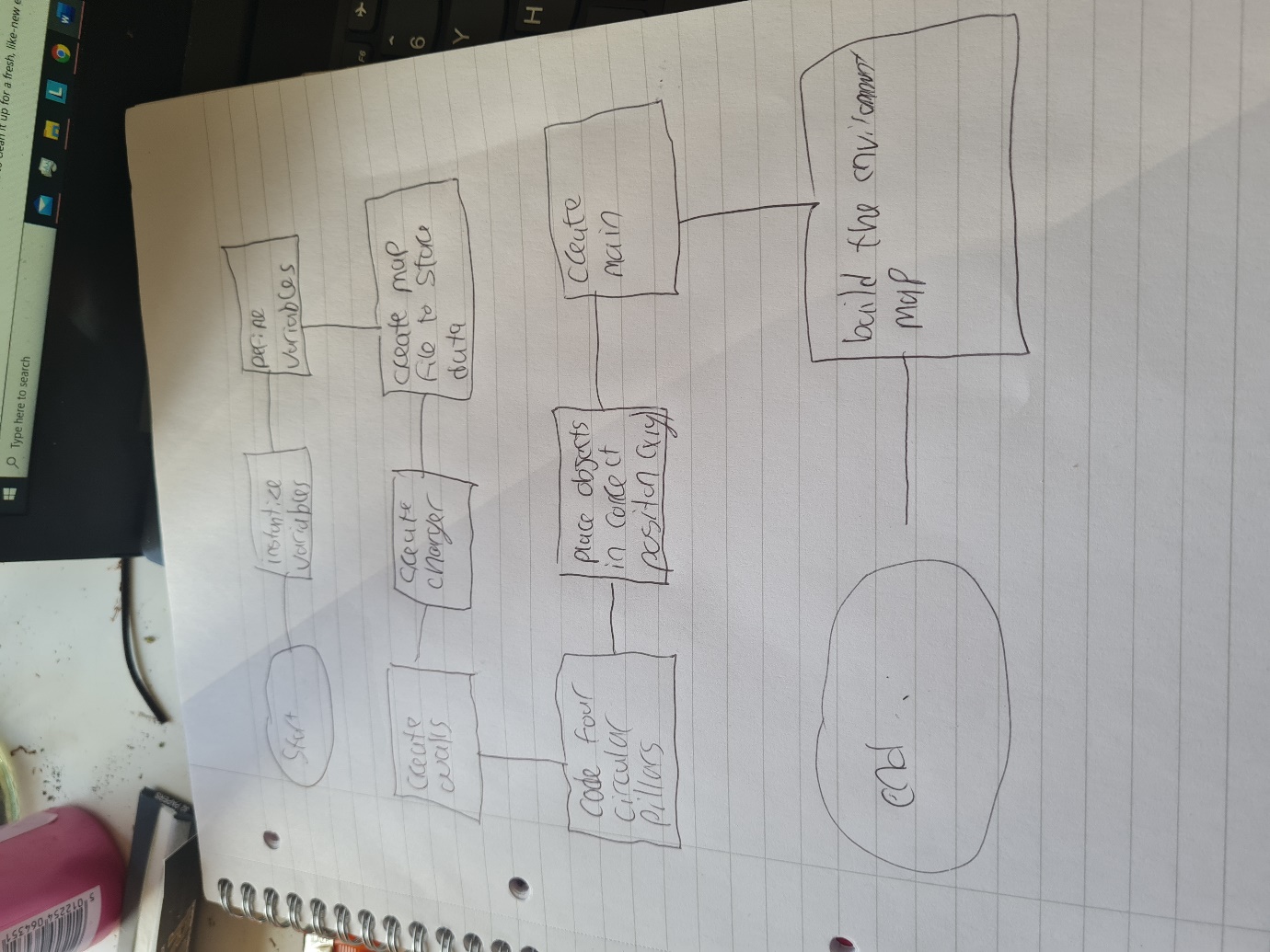
# Task 2



Task 2 required a robot environment map to be built, which contained four pillars and a charge station, these were surrounded by four walls that also had to be created. In order to create this map variables must be instantized and assigned the correct values, for instance looking at the picture provided the wall length for this map is 2.5, so it is a 2.5x2.5 square map, the variable for wall length will then be 2.5. To plot points on this map the x and y positions are important when trying to place a given object, this information is already provided but will need to be stored in an array.

To place a wall on this map a for loop was used to iterate what positions for the line to be made, it incremented in 0.01 until the line reached the end, the second line of the for loop contained the positions of x and y of where the wall line will be plotted.

To create the four circle pillars a for loop was used. To plot a circle pillar, we will need the centre of each circle. By iterating over the array that contains the values that hold the x and y values of the circle pillars the values can then be inputted into the formula and plotted.



## Flowchart of task 2

# Appendix

## Task 1

#include<math.h>

#include<iostream>

#include<fstream>

#define PI 3.14159265

int W=30, delta\_t=1;

const int SIZE=200;

double vl=5, vr=7;

double x\_cur[SIZE]={30}, y\_cur[SIZE]={30}, theta\_cur[SIZE]={PI/4};

int robot\_kinematics(double left\_vel, double right\_vel){

for(int i=0; i<SIZE; i++){

x\_cur[i+1] = x\_cur[i]+((left\_vel+right\_vel)/2)\*cos(theta\_cur[i])\*delta\_t;

y\_cur[i+1] = y\_cur[i]+((left\_vel+right\_vel)/2)\*sin(theta\_cur[i])\*delta\_t;

theta\_cur[i+1] = theta\_cur[i]+((right\_vel-+left\_vel)/W)\*delta\_t;

}

return 0;

}

int main(int argc, char \*\*argv)

{

FILE \*fp;

int i;

fp = fopen ("trajectory2","w");

robot\_kinematics(vl, vr);

fprintf(fp, "%d, %d \n", 30, 30);

for(i=1; i<SIZE; i++){

fprintf(fp, "%f, %f \n", x\_cur[i+1], y\_cur[i+1]);

}

fclose(fp);

return 0;

}

## Task 2

#include<iostream>

#include<fstream>

#include<math.h>

using namespace std;

#define PI 3.14159265

int build\_environment\_map(){

// To initialize all the parameters by using data in the figure above.

double x0[5]={0.8, 0.8, 1.7, 2.2, 2.25};

double y0[5]={1.5, 2.25, 1.0, 1.3, 0.25};

double r=0.125, wallLength=2.5;

double RD=PI /180;

// Open a data file “EnvironmentMap.txt” for storing walls, pillars and charger

ofstream map;

map.open("EnvironmentMap.txt");

// The following code is to create a charger based on its centre.

for (int i=0; i<360; i=i+20) {

map<<x0[4]+r\*cos(i\*RD)<<' '<<y0[4]+r\*sin(i\*RD)<<endl;

}

// For your reference, the following code is used to generate one wall.

for (double j=0 ; j<wallLength; j=j+0.01){

map << j << ' ' << 0 << endl;}

for (double k=0 ; k<wallLength; k=k+0.01){

map << k << ' ' << 2.5 << endl;}

for (double s=0 ; s<wallLength; s=s+0.01){

map << 0 << ' ' << s << endl;}

for (double s=0 ; s<wallLength; s=s+0.01){

map << 2.5 << ' ' << s << endl;}

// put your code here to generate four circle pillars with the given centres.

for (int i=0; i<360; i=i+20) {

map<<x0[3]+r\*cos(i\*RD)<<' '<<y0[3]+r\*sin(i\*RD)<<endl;

}

for (int i=0; i<360; i=i+20) {

map<<x0[2]+r\*cos(i\*RD)<<' '<<y0[2]+r\*sin(i\*RD)<<endl;

}

for (int i=0; i<360; i=i+20) {

map<<x0[1]+r\*cos(i\*RD)<<' '<<y0[1]+r\*sin(i\*RD)<<endl;

}

for (int i=0; i<360; i=i+20) {

map<<x0[0]+r\*cos(i\*RD)<<' '<<y0[0]+r\*sin(i\*RD)<<endl;

}

// put your code here to generate other three walls.

map.close();

return 0;

}

// add the main function into your code for compilation

int main(int argc, char \*\*argv)

{

int i;

i = build\_environment\_map();

return 0;

}