```
//Variables for Ultrasonic pins & left & right IR LED

const int trigPin = 13;

const int echoPin = 12;

long duration;

int distance;

int LIR,RIR;


//Declaring i/p & o/p pins

void setup()

{

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(A0, INPUT);

pinMode(A1, INPUT);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(9600);

}
```

```
//Basic Movement Controls Declaration Start
void forward()
{
digitalWrite(5, HIGH);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
}

void right()
{
digitalWrite(5, LOW);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
}

void left()
{
```

```
digitalWrite(5, HIGH);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, LOW);
}

void clockwise(){
 digitalWrite(5, HIGH);
 digitalWrite(6, LOW);
 digitalWrite(7, HIGH);
 digitalWrite(8, LOW);
}

void anticlockwise(){
 digitalWrite(5, LOW);
 digitalWrite(6, HIGH);
 digitalWrite(7, LOW);
 digitalWrite(8, HIGH);
}
```

```
void Stop()
{
digitalWrite(5, LOW);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, LOW);
delay(2000);
}
//Basic Movement Controls Declaration End


void loop(){
  IR();
  obstacle();
}


//Code for Line Follower
int count = 0;
void IR(){
```

```
LIR = digitalRead(A0);

RIR = digitalRead(A1);


if(LIR==1 && RIR==1)                //When buggy is on the black line

forward();


if(LIR==0 && RIR==1)                //When black line is under the left IR LED

left();


if(LIR==1 && RIR==0)                //When black line is under the right IR LED

right();


//Near Junction or black line(park area) when both IR LEDs are above black lines

if(LIR==0 && RIR==0){

count++;


//First, Sixth & Seventh time the buggy moves forward ignoring the black line

if (count==1 || count==6 || count==7){

 do{
   LIR = digitalRead(A0);

   RIR = digitalRead(A1);

   forward();

   }while(LIR==0 && RIR==0);

}


//The first junction, we move left to stay on the track (Loop Starts)

if (count==2){

 do{

   LIR = digitalRead(A0);

   RIR = digitalRead(A1);

   left();

   delay(200);

   }while(LIR==0 && RIR==0);

}


//The second junction, we move right (to complete 1 loop)

if (count==4){

 do{

   LIR = digitalRead(A0);
   RIR = digitalRead(A1);
   right();
   }while(LIR==0 && RIR==0);
}

//The third junction we move left (towards parking)
if (count==5){
 do{
   LIR = digitalRead(A0);
   RIR = digitalRead(A1);
   left();
   delay(200);
   }while(LIR==0 && RIR==0);
}

//We will park where we started
if (count==8){
 Stop();
 delay(300000);                    //We stop the buggy
}
```

```
 }
}


void obstacle(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  if(distance<=20){
    Stop();
    delay(200);
  }
```

```
}
```