

**Practical No :-1**

**Aim:-**Write a program to create a robot  
i) With gear ii) Without gear  
and move it forward, left, right .

**Program:-**

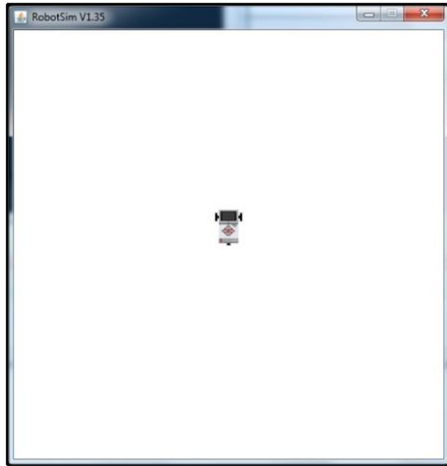
**i ) With gear :-**

**Movewithgear.java**

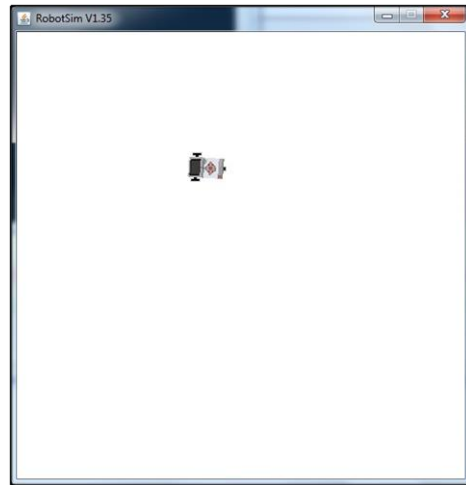
```
import ch.aplu.robotsim.*;
class Movewithgear
{
    Movewithgear()
    {
        NxtRobot robot = new NxtRobot();
        Gear gear = new Gear();
        robot.addPart(gear);
        gear.forward(2000);
        gear.setSpeed(30);
        gear.left(480);
        gear.forward(2000);
        gear.right(480);
        gear.forward();
        robot.exit();
    }
    public static void main(String[] args)
    {
        new Movewithgear();
    }
}
```

**Output :-**

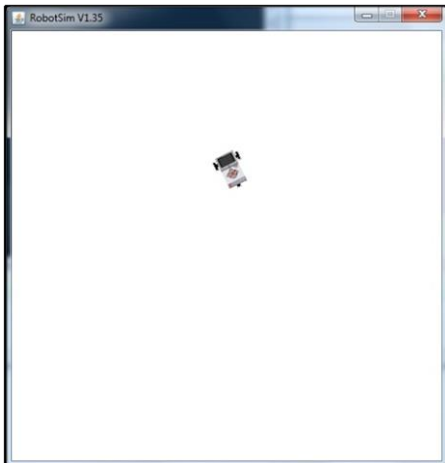
**Initial state :-**



**Move Forward:**



**Turn left :-**

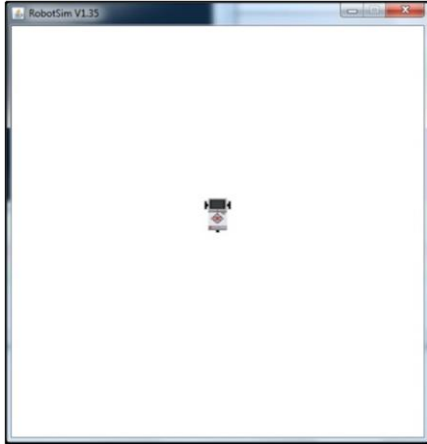


**ii ) Without gear :-****Movewithoutgears.java**

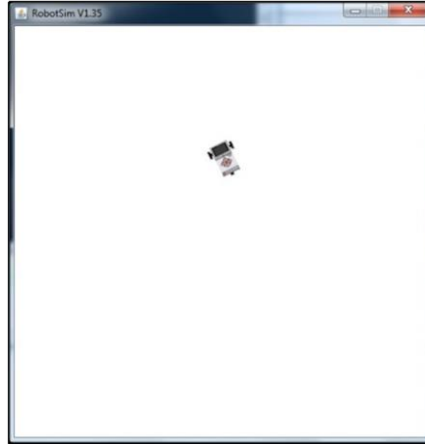
```
import ch.aplu.robotsim.*;
class Movewithoutgear
{
    Movewithoutgear()
    {
        TurtleRobot robot = new TurtleRobot();
        robot.forward(100);
        robot.left(90);
        robot.forward(50);
        robot.right(90);
        robot.forward(50);
        robot.exit();
    }
    public static void main(String[] args)
    {
        new Movewithoutgear();
    }
}
```

**Output :-**

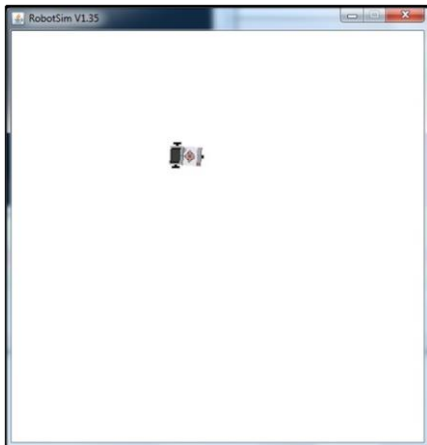
**Initial state :-**



**Turn left :-**



**Move Forward :-**



**Practical No :-2**

**Aim:-** Write a program to create a robot with two motor and move it forward, left and right .

**Program :-****MovewithMotors.java**

```
import ch.aplu.robotsim.*;
public class Movewithmotors
{
    public Movewithmotors()
    {
        NxtRobot robot = new NxtRobot();
        Motor motA = new Motor(MotorPort.A);
        Motor motB = new Motor(MotorPort.B);
        robot.addPart(motA);
        robot.addPart(motB);

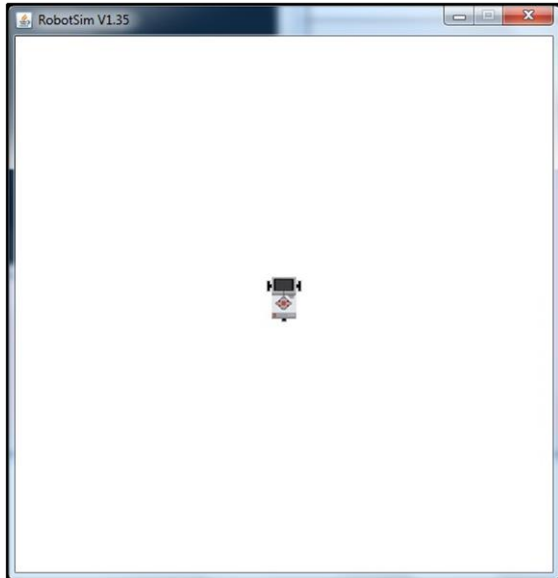
        motA.forward();
        motB.forward();
        Tools.delay(2000);

        motA.stop();
        Tools.delay(1050);
        motA.forward();
        Tools.delay(2000);

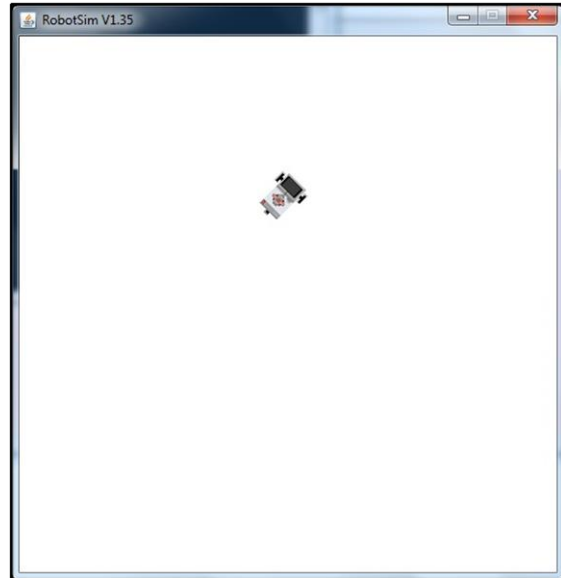
        motB.stop();
        Tools.delay(1050);
        motB.forward();
        Tools.delay(2000);
        robot.exit();
    }
    public static void main(String[]args)
    {
        new Movewithmotors();
    }
}
```

**Output :-**

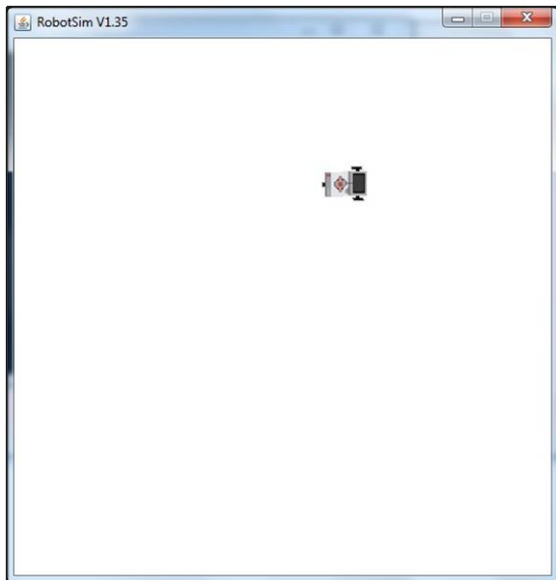
**Initial state :-**



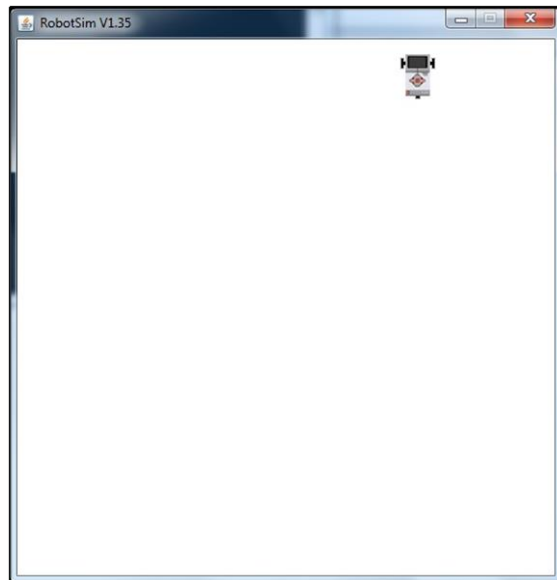
**Turn right :-**



**Move Forward :-**



**Final state :-**



**Practical No :-3**

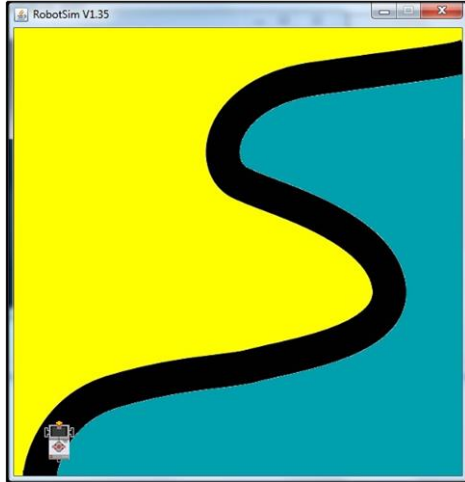
**Aim** :-Write a program to create a robot with light sensors to follow a line .

**Program** :-**LineFollower.java**

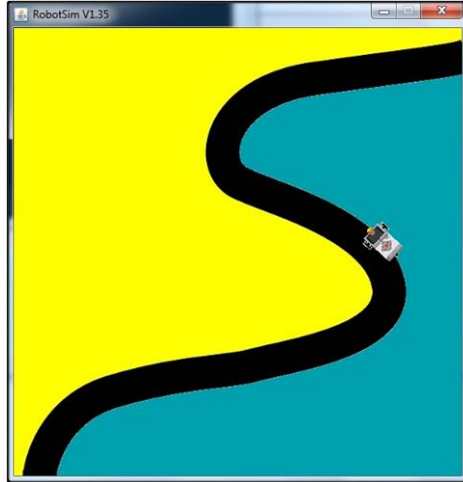
```
import ch.aplu.robotsim.*;
class LineFollower
{
    LineFollower()
    {
        LegoRobot robot = new LegoRobot();
        Gear gear = new Gear();
        LightSensor ls = new LightSensor(SensorPort.S3);
        robot.addPart(gear);
        gear.setSpeed(50);
        robot.addPart(ls);
        while(true)
        {
            int v = ls.getValue();
            if(v<100)
                gear.forward();
            if(v>300 && v<750)
                gear.leftArc(0.05);
            if(v>800)
                gear.rightArc(0.05);
        }
    }
    public static void main(String[]args)
    {
        new LineFollower();
    }
    static
    {
        RobotContext.setStartPosition(50,490);
        RobotContext.setStartDirection(-90);
        RobotContext.useBackground("sprites/road.gif");
    }
}
```

**Output :-**

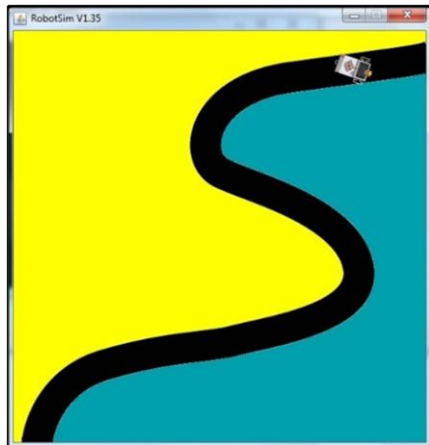
**Initial state :-**



**Intermediate state :-**



**Final state :-**





**Practical No :-4**

**Aim** :-Write a program to create a robot that does a circle using 2 motors .

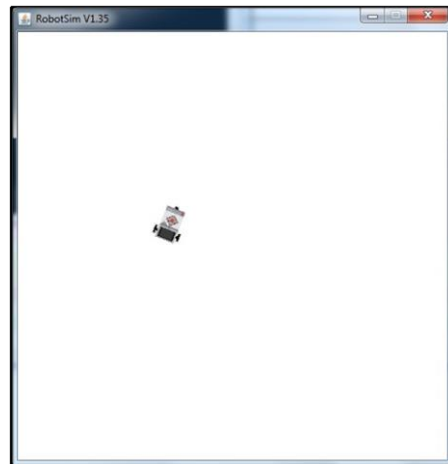
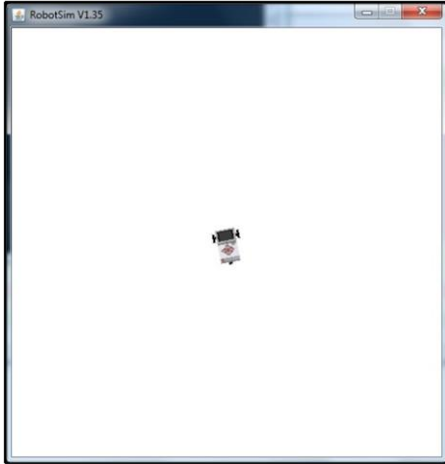
**Program** :-

**Circlem.java**

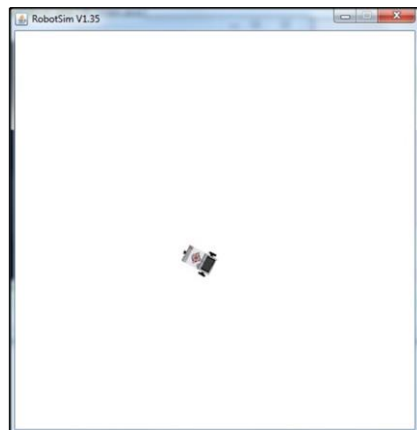
```
import ch.aplu.robotsim.*;
class Circlem
{
    Circlem()
    {
        NxtRobot robot=new NxtRobot();
        Gear gear=new Gear();
        robot.addPart(gear);
        gear.setSpeed(200);
        gear.leftArc(0.2,7000);
        gear.rightArc(0.2);
        Tools.delay(5000);
        robot.exit();
    }
    public static void main(String[]args)
    {
        new Circlem();
    }
}
```

**Output :-**

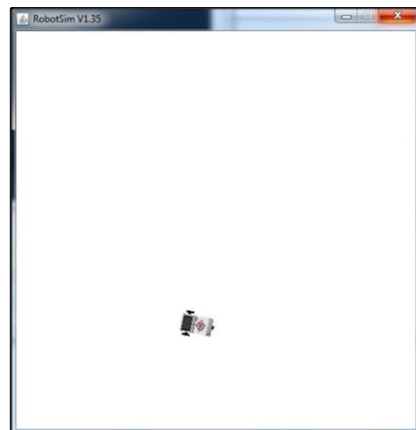
**Initial state :-**



**Intermediate state :-**



**Final state :-**



**Practical No :-5**

**Aim** :-Write a program to create a path following robot .

**Program** :-**PathFinder.java**

```
import ch.aplu.robotsim.*;
public class PathFinder
{
    public PathFinder()
    {
        NxtRobot robot = new NxtRobot();
        Gear gear = new Gear();
        LightSensor ls1 = new LightSensor(SensorPort.S1);
        LightSensor ls2 = new LightSensor(SensorPort.S2);
        robot.addPart(gear);
        robot.addPart(ls1);
        robot.addPart(ls2);
        gear.forward();
        while (true)
        {
            int rightValue = ls1.getValue();
            int leftValue = ls2.getValue();
            int d = rightValue - leftValue;
            if(d > 100)
                gear.rightArc(0.1);
            if(d < -100)
                gear.leftArc(0.1);
            if(d > -100 && d < 100 && rightValue > 500)
                gear.forward();
        }
    }
    public static void main(String[] args)
    {
        new PathFinder();
    }
    static
    {
        NxtContext.setStartPosition(250, 490);
        NxtContext.setStartDirection(-90);
        NxtContext.useBackground("sprites/path.gif");
    }
}
```

**Output :-**

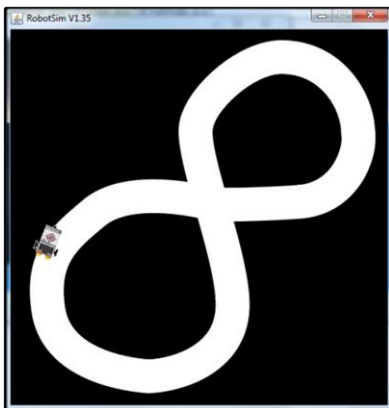
**Initial state :-**



**Intermediate state :-**



**Final State:-**



**Practical No :-6**

**Aim** :-Write a program to implement Breadth First Search(BFS) algorithm for a given standard problem.

**Program :-****breadthfirst.java**

```
import java.util.*;
import java.io.*;
public class breadthfirst
{
    ArrayList arr = new ArrayList();
    String str[] = new String[2];
    String path[] = new String[20];
    int i,j,k=0;
    public breadthfirst()
    {
        arr.add("Dadar");
        arr.add("Kalyan");
        arr.add("Jogeshwari");
        arr.add("Malad");
        arr.add("Kandivali");
        arr.add("Borivali");
        arr.add("Goregaon");
        arr.add("Andheri");
        arr.add("Virar");
        arr.add("Bandra");
        arr.add("Thane");
        arr.add("Bhandup");
        arr.add("Mulund");
        arr.add("Mahim");
        arr.add("Chembur");
    }
    public void breadth()
    {
        if(arr.isEmpty())
            System.out.println("Empty");
        for(i=0;i<20;i++)
            path[i]="";
        str[0]="";
        str[1]="";
        System.out.println("\t\tDadar\n");
        System.out.println("Kalyan\t\tJogeshwari\t\tMalad\n");
        System.out.println("\t\tKandivaliBorivali\tGoregaon\n");
    }
}
```

```
System.out.println("\t\tAndheriVirarBandra\t Thane\n");
System.out.println("\tBhandupMulundMahim\t\tChembur\n");
try
{
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter the initial node");
    str[0]=br.readLine();
    System.out.println("Enter the goal");
    str[1]=br.readLine();
    if(arr.contains(str[1]))
    {
        System.out.println("Goal is found");
        for(j=arr.indexOf(str[0]);j<=arr.indexOf(str[1]);j++)
        {
            path[k]=arr.get(j).toString();
            k++;
        }
    }
}
catch(IOException e)
{}
System.out.print("Path is: ");
for(j=0;j<k;j++)
{
    System.out.print(path[j]);
    if(j!=k-1)
        System.out.print("-->");
}
}
public static void main(String arg[])
{
    breadthfirst b = new breadthfirst();
    b.breadth();
}
}
```

**Output :-**

```
C:\WINDOWS\system32\cmd.exe

          Dadar

Kalyan          Jogeshwari          Malad

          KandivaliBorivali          Goregaon

          AndheriVirarBandra          Thane

          BhandupMulundMahim          Chembur

Enter the initial node
Dadar
Enter the goal
Goregaon
Goal is found
Path is:  Dadar-->Kalyan-->Jogeshwari-->Malad-->Kandivali-->Borivali-->Goregaon
D:\rr\javaq>Pause
Press any key to continue . . . ■
```

**Practical No :-7**

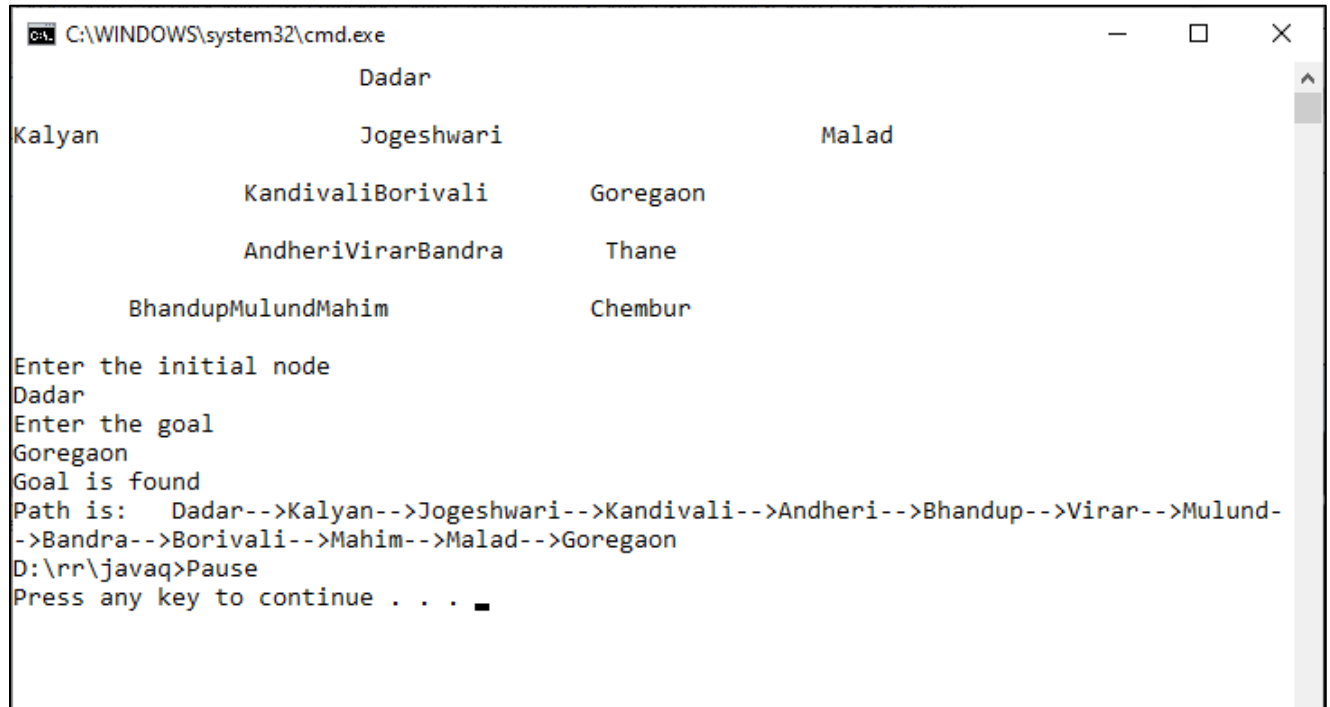
**Aim** :-Write a program to implement Depth First Search(DFS) algorithm for a given standard problem.

**Program :-****depthfirst.java**

```
import java.util.*;
import java.io.*;
public class depthfirst
{
    ArrayList arr = new ArrayList();
    String str[] = new String[2];
    String path[] = new String[20];
    int i,j,k=0;
    public depthfirst()
    {
        arr.add("Dadar");
        arr.add("Kalyan");
        arr.add("Jogeshwari");
        arr.add("Kandivali");
        arr.add("Andheri");
        arr.add("Bhandup");
        arr.add("Virar");
        arr.add("Mulund");
        arr.add("Bandra");
        arr.add("Borivali");
        arr.add("Mahim");
        arr.add("Malad");
        arr.add("Goregaon");
        arr.add("Thane");
        arr.add("Chembur");
    }
    public void depthsearch()
    {
        if(arr.isEmpty())
            System.out.println("Empty");
        for(i=0;i<20;i++)
            path[i]="";
        str[0]="";
        str[1]="";
        System.out.println("\t\tDadar\n");
        System.out.println("Kalyan\t\tJogeshwari\t\tMalad\n");
        System.out.println("\t\tKandivaliBorivali\tGoregaon\n");
```



```
System.out.println("\t\tAndheriVirarBandra\t Thane\n");
System.out.println("\t\tBhandupMulundMahim\t\tChembur\n");
try
{
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter the initial node");
    str[0]=br.readLine();
    System.out.println("Enter the goal");
    str[1]=br.readLine();
    if(arr.contains(str[1]))
    {
        System.out.println("Goal is found");
        for(j=arr.indexOf(str[0]);j<=arr.indexOf(str[1]);j++)
        {
            path[k]=arr.get(j).toString();
            k++;
        }
    }
}
catch(IOException e)
{}
System.out.print("Path is: ");
for(j=0;j<k;j++)
{
    System.out.print(path[j]);
    if(j!=k-1)
        System.out.print("-->");
}
}
public static void main(String arg[])
{
    depthfirst b = new depthfirst();
    b.depthsearch();
}
}
```

**Output :-**

```
C:\WINDOWS\system32\cmd.exe

          Dadar

Kalyan          Jogeshwari          Malad

          KandivaliBorivali      Goregaon

          AndheriVirarBandra      Thane

          BhandupMulundMahim      Chembur

Enter the initial node
Dadar
Enter the goal
Goregaon
Goal is found
Path is:  Dadar-->Kalyan-->Jogeshwari-->Kandivali-->Andheri-->Bhandup-->Virar-->Mulund-
->Bandra-->Borivali-->Mahim-->Malad-->Goregaon
D:\rr\javaq>Pause
Press any key to continue . . .
```

**Practical No :- 8**

**Aim** :-Write a program to implement A\* search algorithm for a given standard problem .

**Program** :-**Astar.java**

```
import java.util.*;
import java.io.*;
class Astar
{
    String str[]=new String[2];
    String s1,s2="";
    int no,i,j,min=1,temp,k=0;
    String path[] = new String[50];
    public Astar()throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter the initial node:-\n");
        str[0]=br.readLine();
        path[k]=str[0];
        System.out.print("Enter the Goal Node:-\n");
        str[1]=br.readLine();
        System.out.println("\t\tDadar\n");
        System.out.println("Kalyan\t\tJogeshwari\t\tMalad\n");
        System.out.println("\tKurla\tKandivaliBorivali\tGoregaon\n");
        System.out.println("\tAndheriVirarBandra\tThane\n");
        System.out.println("BhandupMulundMahim\t\tChembur\n");
        while(!str[0].equals(str[1]))
        {
            System.out.print(s2);
            System.out.println("Enter the no. of node of" + " " + str[0] );
            s1=br.readLine();
            no=Integer.parseInt(s1);
            String st[][] = new String[no][3];
            int arr[] = new int[no];
            System.out.println("Enter the successor nodes of" + " " + str[0]+" " +"With their straight line distance to goal node h(n) and path cost from start node i.e. h(n) ");
            for(i=0;i<no;i++)
            {
                for(j=0;j<3;j++)
                {
                    st[i][j]=br.readLine();
                }
            }
        }
    }
}
```

```
for(i=0;i<no;i++)
{
    arr[i]=Integer.parseInt(st[i][1])+Integer.parseInt(st[i][2]);
}
for(i=0;i<no;i++)
{
    for(j=i+1;j<no;j++)
    {
        if(arr[i]<arr[j])
        {
            temp=arr[i];
            arr[j]=arr[i];
            arr[i]=temp;
        }
        else
        {
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
        }
    }
}
min=arr[0];
for(i=0;i<no;i++)
{
    if((Integer.parseInt(st[i][1])+Integer.parseInt(st[i][2]))==min)
    str[0]=st[i][0];
    s2="Node"+" "+str[0]+" "+"has smallest value therefore ";
}
k++;
path[k]=str[0];
}
System.out.println("The path from initial node to goal node is:");
for(i=0;i<k;i++)
System.out.print(path[i]+"--->");
System.out.print(str[1]);
}
public static void main(String arg[])throws IOException
{
    Astar a = new Astar();
}
}
```

**Output :-**

```

C:\WINDOWS\system32\cmd.exe
Enter the initial node:-
Dadar
Enter the Goal Node:-
Bandra

          Dadar
Kalyan          Jogeshwari          Malad
      Kurla  KandivaliBorivali      Goregaon
      AndheriVirarBandra      Thane
BhandupMulundMahim          Chembur

Enter the no. of node of Dadar
3
Enter the successor nodes of Dadar With their straight line distance to goal node h(n) and path
cost from start node i.e. h(n)
Jogeshwari
253
140
Malad
329
118
Kalyan
374
75
Node Jogeshwari has smallest value therefore Enter the no. of node of Jogeshwari
4
Enter the successor nodes of Jogeshwari With their straight line distance to goal node h(n) and
path cost from start node i.e. h(n)
Dadar
366
280
Borivali
176
239
Kurla
380
291
Kandivali
193
220

Node Kandivali has smallest value therefore Enter the no. of node of Kandivali
3
Enter the successor nodes of Kandivali With their straight line distance to goal node h(n) and
path cost from start node i.e. h(n)
Andheri
366
160
Virar
317
100
Jogeshwari
253
300
Node Virar has smallest value therefore Enter the no. of node of Virar
3
Enter the successor nodes of Virar With their straight line distance to goal node h(n) and path
cost from start node i.e. h(n)
Bandra
0
418
Andheri
160
455
Kandivali
193
414
The path from initial node to goal node is:
Dadar--->Jogeshwari--->Kandivali--->Virar--->Bandra
D:\rr\javaq>Pause
Press any key to continue . . .

```