

Работа победителя/призера заключительного этапа
командной инженерной олимпиады школьников
Олимпиада Национальной технологической инициативы

Профиль «Интеллектуальные робототехнические системы»

Воронов Роман Витальевич

Класс: 9

Город: Новосибирск

Школа: МБОУ ЛИТ

Регион: Новосибирская
область

Уникальный номер участника: 12

**Команда на заключительном
этапе:** Ravenator

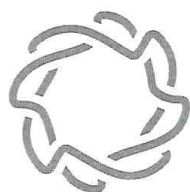
Параллель: 9 класс

Результаты заключительного этапа:

№	индивидуальная			Командная часть				результат
	Математика	информатика	итого	1 день	2 день	3 день	общий	
12	21	5	26				92	118

Индивидуальная часть

Персональный лист участника с номером 12:



Олимпиада НТИ

ФИО Воронов Роман Витальевич

Город Новосибирск

Школа № МБОУ ЛИТ

Математика

Лист 1:

Командная инженерная олимпиада «Олимпиада НТИ»

Направление ИРС

Предмет МАТЕМАТИКА

Номер участника 353892

1	2	3
20	5	0

N2

a) Дано:

$$AB = 1 \text{ м}$$

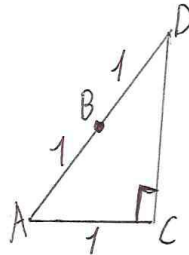
$$BD = 1 \text{ м}$$

$$AC = 1 \text{ м}$$

$$\angle ACD = 90^\circ$$

DC

Решение:



$$DC = \sqrt{(AB+BD)^2 - AC^2}$$

$$DC = \sqrt{2^2 - 1^2} = \sqrt{3} \text{ (м)}$$

$$\sqrt{3} \approx 1,732 \text{ (м)} \quad \leftarrow \text{66}$$

N3

а)

$$1) 3809 - 1625 = 2184 \text{ (первое изменение } p_t)$$

$$2) 6929 - 3809 = 3120 \text{ (второе изменение } p_t)$$

$$3) 3120 - 2184 = 936 \text{ (скорость изменений } p_t)$$

$$4) 3120 + 936 = 4056 \text{ (третье изменение } p_t)$$

$$5) 6929 + 4056 = 10985 \text{ (значение } p_t \text{ при } t=3c)$$

$$6) 4409 - 3625 = 884 \text{ (первое изменение } q_t)$$

$$7) 6529 - 4409 = 2120 \text{ (второе изменение } q_t)$$

$$8) 2120 - 884 = 1236 \text{ (скорость изменений } q_t)$$

$$9) 2120 + 1236 = 3356 \text{ (третье изменение } q_t)$$

$$10) 6529 + 3356 = 9885 \text{ (значение } q_t \text{ при } t=3c)$$

Ответ: $p_3 = 10985$; $q_3 = 9885$

Информатика

Задача A0:

```
var n,m:integer;
fir:array[1..500,1..2] of int64;
sec:array[1..500,1..2] of int64;

function peres(x1,y1,x2,y2,x3,y3,x4,y4:int64):boolean;
begin
if ((max(x1,x2)> min(x3,x4)) and (min(x1,x2)< max(x3,x4)) and (max(y1,y2)> min(y3,y4)) and
(min(y1,y2)< max(y3,y4))) then
result:=true
else
result:=false;
if (min(x1,x2)> min(x3,x4)) and (max(x1,x2) > max(x3,x4)) then
result:=false;
if (min(x1,x2)< min(x3,x4)) and (max(x1,x2) < max(x3,x4)) then
result:=false;
if (min(y1,y2)> min(y3,y4)) and (max(y1,y2) > max(y3,y4)) then
result:=false;
if (min(y1,y2)< min(y3,y4)) and (max(y1,y2) < max(y3,y4)) then
result:=false;

if ((x1=x2) and ((x1 = min(x3,x4)) or (x1 = max(x3,x4)))) then
result:=true;
if ((x3=x4) and ((x3 = min(x1,x2)) or (x3 = max(x1,x2)))) then
result:=true;
if ((y1=y2) and ((y1 = min(y3,y4)) or (y1 = max(y3,y4)))) then
result:=true;
if ((y3=y4) and ((y3 = min(y1,y2)) or (y3 = max(y1,y2)))) then
result:=true;
if ((x1=x3)and(y1=y3)) or ((x1=x4)and(y1=y4)) or ((x2=x3)and(y2=y3)) or
((x2=x4)and(y2=y4)) then
result:=true;

end;

begin
```

```
read(n,m);
for var i:=1 to n do
begin
  read(fir[i,1]);
  read(fir[i,2]);
end;
for var i:=1 to m do
begin
  read(sec[i,1]);
  read(sec[i,2]);
end;
if peres(fir[1,1],fir[1,2],fir[2,1],fir[2,2],sec[1,1],sec[1,2],sec[2,1],sec[2,2]) then
writeln('Yes')
else
writeln('No');
end.
```

Командная часть

Результаты были получены в рамках выступления команды: Ravenator



Личный состав команды:

- Воронов Роман Витальевич,
- Воронов Артем Витальевич.

Протокол выполнения заданий:

Первый этап:

№ задания	Критерий	Возможные баллы	Полученные баллы
1	Симулятор: программа управления вывела оптимальный маршрут верно для всех проверочных полигонах	4x1	4
2	Симулятор: робот проехал из точки старта в точку финиша на всех проверочных полигонах	8x1	8
3	Робот покинул сектор старта	4x2	8
4	Робот проехал половину от всего количества секторов, находящихся на оптимальном пути от сектора старта до сектора финиша	8x2	8
5	Робот проехал от старта до финиша, остановился и вывел на экран finish	16x2	16
6	Маршрут, по которому проехал робот от старта до финиша является оптимальным.	12x2	12
7	Вместо finish на экран робота выведено верное количество перекрытых справа от робота секторов	4x2	4
Дополнительные баллы		32	0
<i>Всего за этап</i>		132	60

Второй этап:

№ задания	Критерий	Возможные баллы	Полученные баллы
8	Симулятор: робот проехал из точки старта в точку финиша на всех проверочных полигонах	12x1	0
9	Робот распознал верно ARTag метку, которая установлен прямо перед камерой, и вывел на экран число, закодированное меткой	4x2	0
10	Робот располагается за два сектора до сектора распределения задач, при этом он расположен перпендикулярно сектору запуска. Робот доехал до сектора распределения задач, остановился, издал звуковой сигнал, вывел на экран верное значение первой по ходу движения ARTag метки и продолжил движение через 10 секунд	4x2	0
11	Робот располагается за два сектора до сектора распределения задач, при этом он расположен перпендикулярно сектору запуска. Робот доехал до сектора распределения задач, остановился, издал звуковой сигнал, вывел на экран верное значение второй по ходу движения ARTag метки	4x2	0
12	Робот проехал от старта до сектора распределения задач, остановился, издал звуковой сигнал, вывел на экран верные координаты сектора финиша и продолжил движение через 10 секунд	28x2	0

13	Робот остановился в секторе finish и вывел finish	4x2	0
Дополнительные баллы		32	0
<i>Всего за этап</i>		132	0

Третий этап:

№ задания	Критерий	Возможные баллы	Полученные баллы
14	Симулятор: робот проехал из точки старта в точку finish и верно вывел количество недоступных секторов	24x1	24
15	Робот остановился в секторе соседнем от сектора с первым по ходу движения завалом и воспроизвел звук, робот не касался элементов завалов до момента остановки рядом с данным завалом	4x2	4
16	После остановки робота в секторе с первым по ходу движения завалом на экран верно выведено минимально возможное количество секторов недоступных для посещения, согласно изменениям на карте, исследованным к моменту остановки.	4x2	4
17	Робот остановился в секторе соседнем от сектора со вторым по ходу движения завалом и воспроизвел звук, робот не касался элементов завалов во время выполнения задания до момента остановки рядом с данным завалом	4x2	0
18	После остановки робота возле сектора со вторым по ходу движения завалом на экран верно выведено минимально возможное количество секторов недоступных для посещения, согласно изменениям на карте, исследованным к моменту остановки.	4x2	0
19	Робот доехал до сектора распределения задач, остановился, издал звуковой сигнал, вывел на экран верные координаты сектора finish и продолжил движение через 10 секунд, до момента остановки в секторе распределения задач робот не касался элементов любых завалов	4x2	0
20	Робот остановился в секторе finish и вывел finish, до момента остановки в секторе finish робот не касался элементов любых завалов	12x2	0
21	Вместо finish на экран робота верно выведено количество недоступных для посещения секторов	8x2	0
Дополнительные баллы		32	0
<i>Всего за этап</i>		136	32

Программа команды для решения задачи первого этапа.

Решение задачи в Симуляторе:

```
function Kletka(x,y,left,up,right,down)
{
    this.x=x;
    this.y=y;
    this.left=left;
    this.up=up;
    this.right=right;
    this.down=down;
}

function Shir(y1,x1,V,y2,x2)
{
    var S= x1+1 + y1*8;
    var F= x2+1 + y2*8;
    napr[S]=V;
    add(S);
    while (true) {
        ost = take();
        if (ost == F) {
            break;
        };
        if (sector[ost].left == 1) {
            if (napr[ost] == 0) {
                K = put[ost]+"F";
            };
            if (napr[ost] == 1) {
                K = put[ost]+"L"+"F";
            };
            if (napr[ost] == 2) {
```

```

        K = put[ost]+"LL"+"F";
};
if (napr[ost] == 3) {
    K = put[ost]+"R"+"F";
};
if (pos[ost-1] == 0) {
    put[ost-1] = K;
    add (ost-1);
    napr[ost-1] = 0;
} else {
    if (K < put[ost-1]) {
        put[ost-1] = K;
        add (ost-1);
        napr[ost-1] = 0;
    }
}
};
if (sector[ost].up == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"R"+"F";
    };
    if (napr[ost] == 1) {
        K = put[ost]+"F";
    };
    if (napr[ost] == 2) {
        K = put[ost]+"L"+"F";
    };
    if (napr[ost] == 3) {
        K = put[ost]+"LL"+"F";
    };
};

```

```

        if (pos[ost+8] == 0) {
            put[ost+8] = K;
            add (ost+8);
            napr[ost+8] = 1;
        } else {
            if (K < put[ost+8]) {
                put[ost+8] = K;
                add (ost+8);
                napr[ost+8] = 1;
            }
        }
    };

    if (sector[ost].right == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"LL"+"F";
        };
        if (napr[ost] == 1) {
            K = put[ost]+"R"+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"F";
        };
        if (napr[ost] == 3) {
            K = put[ost]+"L"+"F";
        };
        if (pos[ost+1] == 0) {
            put[ost+1] = K;
            add (ost+1);
            napr[ost+1] = 2;
        } else {
            if (K < put[ost+1]) {

```

```

        put[ost+1] = K;
        add (ost+1);
        napr[ost+1] = 2;
    }
}

};

if (sector[ost].down == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"L"+"F";
    }
    if (napr[ost] == 1) {
        K = put[ost]+"LL"+"F";
    }
    if (napr[ost] == 2) {
        K = put[ost]+"R"+"F";
    }
    if (napr[ost] == 3) {
        K = put[ost]+"F";
    }
    if (pos[ost-8] == 0) {
        put[ost-8] = K;
        add (ost-8);
        napr[ost-8] = 3;
    } else {
        if (K < put[ost-8]) {
            put[ost-8] = K;
            add (ost-8);
            napr[ost-8] =3;
        }
    }
}

```

```

        }
    }
    return put[F];
}
function add(q) {
    klet[last]=q;
    last=last+1;
    pos[q]=1;
};

function length() {
    return last-fir;
};

function take() {
    fir=fir+1;
    return klet[fir-1];
}

var run = function(length) {
    var leng = length;
    var p =90;
    var ob = 75 * Math.PI;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили енкодеры

    mr=brick.motor(M4);
    ml=brick.motor(M3);
    mr.setPower(p);

```

```

ml.setPower(p); //запустили моторы

var limit = (374/ob)*leng;

var enc = 0;

var k = 2;

var razn = 0;

while (enc < limit) {

    enc=(Math.abs(el.readRawData()) +
Math.abs(er.readRawData())) / 2;

    razn = (Math.abs(el.readRawData()) -
Math.abs(er.readRawData())) *k

    mr.setPower(p+razn);

    ml.setPower(p-razn);

    script.wait(5);

}

mr.powerOff();

ml.powerOff();

script.wait(500);

}

var rotate = function(ar) {

    run(100);

    var arc = ar;

    var k = 117;

    var el = brick.encoder(E3);

    var er = brick.encoder(E4);

    el.reset();

    er.reset(); // обнулили энкодеры

    mr=brick.motor(M4);

    ml=brick.motor(M3);

    var ob = 75 * Math.PI;

```

```

if (arc < 0) {
    var leng = 2*k*Math.PI* -arc / 360;
    mr.setPower(75);
    ml.setPower(-75);
}
else {
    var leng = 2*k*Math.PI* arc / 360;
    mr.setPower(-75);
    ml.setPower(75);
}
var limit = (360/ob)*leng;
var enc = 0;
while (enc < limit) {
    enc=Math.abs(er.readRawData());
    script.wait(5);
}
mr.powerOff();
ml.powerOff();
script.wait(500);
}

```

```

var sector = new Array(64);
sector [1]= new Kletka(0,0 ,0,1,1,0);
sector [2]= new Kletka(1,0 ,1,0,1,0);
sector [3]= new Kletka(2,0 ,1,1,0,0);
sector [4]= new Kletka(3,0 ,0,0,1,0);
sector [5]= new Kletka(4,0 ,1,1,1,0);
sector [6]= new Kletka(5,0 ,1,0,0,0);
sector [7]= new Kletka(6,0 ,0,1,1,0);
sector [8]= new Kletka(7,0 ,1,0,0,0);
sector [9]= new Kletka(0,1 ,0,1,0,1);

```

```
sector [10]= new Kletka(1,1 ,0,0,0,0);
sector [11]= new Kletka(2,1 ,0,1,0,1);
sector [12]= new Kletka(3,1 ,0,0,0,0);
sector [13]= new Kletka(4,1 ,0,1,0,1);
sector [14]= new Kletka(5,1 ,0,0,0,0);
sector [15]= new Kletka(6,1 ,0,1,0,1);
sector [16]= new Kletka(7,1 ,0,0,0,0);
sector [17]= new Kletka(0,2 ,0,1,1,1);
sector [18]= new Kletka(1,2 ,1,0,1,0);
sector [19]= new Kletka(2,2 ,1,0,1,1);
sector [20]= new Kletka(3,2 ,1,0,1,0);
sector [21]= new Kletka(4,2 ,1,1,1,1);
sector [22]= new Kletka(5,2 ,1,0,1,0);
sector [23]= new Kletka(6,2 ,1,1,1,1);
sector [24]= new Kletka(7,2 ,1,1,0,0);
sector [25]= new Kletka(0,3 ,0,1,0,1);
sector [26]= new Kletka(1,3 ,0,0,1,0);
sector [27]= new Kletka(2,3 ,1,1,1,0);
sector [28]= new Kletka(3,3 ,1,0,0,0);
sector [29]= new Kletka(4,3 ,0,1,0,1);
sector [30]= new Kletka(5,3 ,0,0,0,0);
sector [31]= new Kletka(6,3 ,0,0,1,1);
sector [32]= new Kletka(7,3 ,1,1,0,1);
sector [33]= new Kletka(0,4 ,0,1,0,1);
sector [34]= new Kletka(1,4 ,0,0,0,0);
sector [35]= new Kletka(2,4 ,0,1,0,1);
sector [36]= new Kletka(3,4 ,0,0,0,0);
sector [37]= new Kletka(4,4 ,0,1,1,1);
sector [38]= new Kletka(5,4 ,1,1,0,0);
sector [39]= new Kletka(6,4 ,0,0,0,0);
```



```
sector [40]= new Kletka(7,4 ,0,1,0,1);
sector [41]= new Kletka(0,5 ,0,0,1,1);
sector [42]= new Kletka(1,5 ,1,1,1,0);
sector [43]= new Kletka(2,5 ,1,0,1,1);
sector [44]= new Kletka(3,5 ,1,1,1,0);
sector [45]= new Kletka(4,5 ,1,0,1,1);
sector [46]= new Kletka(5,5 ,1,1,1,1);
sector [47]= new Kletka(6,5 ,1,0,0,0);
sector [48]= new Kletka(7,5 ,0,1,0,1);
sector [49]= new Kletka(0,6 ,0,0,0,0);
sector [50]= new Kletka(1,6 ,0,1,0,1);
sector [51]= new Kletka(2,6 ,0,0,0,0);
sector [52]= new Kletka(3,6 ,0,1,0,1);
sector [53]= new Kletka(4,6 ,0,0,0,0);
sector [54]= new Kletka(5,6 ,0,1,0,1);
sector [55]= new Kletka(6,6 ,0,0,0,0);
sector [56]= new Kletka(7,6 ,0,1,0,1);
sector [57]= new Kletka(0,7 ,0,0,1,0);
sector [58]= new Kletka(1,7 ,1,0,0,1);
sector [59]= new Kletka(2,7 ,0,0,1,0);
sector [60]= new Kletka(3,7 ,1,0,1,1);
sector [61]= new Kletka(4,7 ,1,0,0,0);
sector [62]= new Kletka(5,7 ,0,0,1,1);
sector [63]= new Kletka(6,7 ,1,0,1,0);
sector [64]= new Kletka(7,7 ,1,0,0,1);

var klet = new Array();
var pos  = new Array(64);
var put  = new Array(64);
var napr = new Array(64);
var fir =1;
```

```

var last =1;
for(var i = 1;i<65;i++)
{
    pos[i]=0;
};
for(var i = 1;i<65;i++)
{
    put[i]="";
};
for(var i = 1;i<65;i++)
{
    napr[i]=0;
}
var Text = script.readAll("input.txt");
p1 = Text[0].indexOf(" ");
p2 = Text[0].indexOf(" ",p1+1);
p3 = Text[1].indexOf(" ");
c1 = +Text[0].substring(0,p1);
c2 = +Text[0].substring(p1+1,p2);
c3 = +Text[0].substring(p2+1);
c4 = +Text[1].substring(0,p3);
c5 = +Text[1].substring(p3+1);
otv=Shir(c1,c2,c3,c4,c5);

print(otv);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
while (otv.length > 0) {
    sim = otv.substring(0,1);
    otv = otv.substring(1);
}

```

```

    if (sim=="F")
    {
        run(870);
    }
    if (sim=="L")
    {
        rotate(88)
    }
    if (sim=="R")
    {
        rotate(-88)
    }
}

brick.display().addLabel("finish",2,2);

brick.display().redraw();

```

Решение задачи на реальном роботе:

```

function Kletka(x,y,left,up,right,down)
{
    this.x=x;
    this.y=y;
    this.left=left;
    this.up=up;
    this.right=right;
    this.down=down;
}
function Shir(y1,x1,V,y2,x2)
{
    var S= x1+1 + y1*8;
    var F= x2+1 + y2*8;
    napr[S]=V;
    add(S);
    while (true) {
        ost = take();
        if (ost == F) {
            break;
        };
        if (sector[ost].left == 1) {
            if (napr[ost] == 0) {
                K = put[ost]+"F";
            };
        };
    };
}

```

```

        if (napr[ost] == 1) {
            K = put[ost]+"L"+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"LL"+"F";
        };
        if (napr[ost] == 3) {
            K = put[ost]+"R"+"F";
        };
        if (pos[ost-1] == 0) {
            put[ost-1] = K;
            add (ost-1);
            napr[ost-1] = 0;
        } else {
            if (K < put[ost-1]) {
                put[ost-1] = K;
                add (ost-1);
                napr[ost-1] = 0;
            }
        }
    };
    if (sector[ost].up == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"R"+"F";
        };
        if (napr[ost] == 1) {
            K = put[ost]+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"L"+"F";
        };
        if (napr[ost] == 3) {
            K = put[ost]+"LL"+"F";
        };
        if (pos[ost+8] == 0) {
            put[ost+8] = K;
            add (ost+8);
            napr[ost+8] = 1;
        } else {
            if (K < put[ost+8]) {
                put[ost+8] = K;
                add (ost+8);
                napr[ost+8] = 1;
            }
        }
    };
    if (sector[ost].right == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"LL"+"F";
        };
        if (napr[ost] == 1) {
            K = put[ost]+"R"+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"F";
        };
        if (napr[ost] == 3) {
            K = put[ost]+"L"+"F";
        };
        if (pos[ost+1] == 0) {
            put[ost+1] = K;

```

```

        add (ost+1);
        napr[ost+1] = 2;
    } else {
        if (K < put[ost+1]) {
            put[ost+1] = K;
            add (ost+1);
            napr[ost+1] = 2;
        }
    }
};
if (sector[ost].down == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"L"+"F";
    }
    if (napr[ost] == 1) {
        K = put[ost]+"LL"+"F";
    }
    if (napr[ost] == 2) {
        K = put[ost]+"R"+"F";
    }
    if (napr[ost] == 3) {
        K = put[ost]+"F";
    }
    if (pos[ost-8] == 0) {
        put[ost-8] = K;
        add (ost-8);
        napr[ost-8] = 3;
    } else {
        if (K < put[ost-8]) {
            put[ost-8] = K;
            add (ost-8);
            napr[ost-8] =3;
        }
    }
}
}
return put[F];
}
function add(q) {
    klet[last]=q;
    last=last+1;
    pos[q]=1;
};

function length() {
    return last-fir;
};

function take() {
    fir=fir+1;
    return klet[fir-1];
}
var run = function(length) {
    var leng = length;
    var p =90;
    var ob = 75 * Math.PI;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили енкодеры
};

```

```

    mr=brick.motor(M4);
    ml=brick.motor(M3);
    mr.setPower(p);
    ml.setPower(p); //запустили моторы

    var limit = (374/ob)*leng;
    var enc = 0;
    var k = 2;
    var razn = 0;
    while (enc < limit) {
        enc=(Math.abs(el.readRawData()) +
Math.abs(er.readRawData())) / 2;

        razn = (Math.abs(el.readRawData())-
Math.abs(er.readRawData()))*k
        mr.setPower(p+razn);
        ml.setPower(p-razn);
        script.wait(5);
    }
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    script.wait(500);
}
var rotate = function(ar) {
    //run(20);
    var arc = ar;
    var k = 117;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset(); // обнулили энкодеры
    mr=brick.motor(M4);
    ml=brick.motor(M3);
    var ob = 75 * Math.PI;
    if (arc < 0) {
        var leng = 2*k*Math.PI* -arc / 360;
        mr.setPower(75);
        ml.setPower(-75);
    }
    else {
        var leng = 2*k*Math.PI* arc / 360;
        mr.setPower(-75);
        ml.setPower(75);
    }
    var limit = (360/ob)*leng;
    var enc = 0;
    while (enc < limit) {
        enc=Math.abs(er.readRawData());
        script.wait(5);
    }
    mr.powerOff();
    ml.powerOff();
    script.wait(500);
}

var sector = new Array(64);
sector [1]= new Kletka(0,0 ,0,1,1,0);
sector [2]= new Kletka(1,0 ,1,0,1,0);
sector [3]= new Kletka(2,0 ,1,1,0,0);
sector [4]= new Kletka(3,0 ,0,0,1,0);

```

```
sector [5]= new Kletka(4,0 ,1,1,1,0);
sector [6]= new Kletka(5,0 ,1,0,0,0);
sector [7]= new Kletka(6,0 ,0,1,1,0);
sector [8]= new Kletka(7,0 ,1,0,0,0);
sector [9]= new Kletka(0,1 ,0,1,0,1);
sector [10]= new Kletka(1,1 ,0,0,0,0);
sector [11]= new Kletka(2,1 ,0,1,0,1);
sector [12]= new Kletka(3,1 ,0,0,0,0);
sector [13]= new Kletka(4,1 ,0,1,0,1);
sector [14]= new Kletka(5,1 ,0,0,0,0);
sector [15]= new Kletka(6,1 ,0,1,0,1);
sector [16]= new Kletka(7,1 ,0,0,0,0);
sector [17]= new Kletka(0,2 ,0,1,1,1);
sector [18]= new Kletka(1,2 ,1,0,1,0);
sector [19]= new Kletka(2,2 ,1,0,1,1);
sector [20]= new Kletka(3,2 ,1,0,1,0);
sector [21]= new Kletka(4,2 ,1,1,1,1);
sector [22]= new Kletka(5,2 ,1,0,1,0);
sector [23]= new Kletka(6,2 ,1,1,1,1);
sector [24]= new Kletka(7,2 ,1,1,0,0);
sector [25]= new Kletka(0,3 ,0,1,0,1);
sector [26]= new Kletka(1,3 ,0,0,1,0);
sector [27]= new Kletka(2,3 ,1,1,1,0);
sector [28]= new Kletka(3,3 ,1,0,0,0);
sector [29]= new Kletka(4,3 ,0,1,0,1);
sector [30]= new Kletka(5,3 ,0,0,0,0);
sector [31]= new Kletka(6,3 ,0,0,1,1);
sector [32]= new Kletka(7,3 ,1,1,0,1);
sector [33]= new Kletka(0,4 ,0,1,0,1);
sector [34]= new Kletka(1,4 ,0,0,0,0);
sector [35]= new Kletka(2,4 ,0,1,0,1);
sector [36]= new Kletka(3,4 ,0,0,0,0);
sector [37]= new Kletka(4,4 ,0,1,1,1);
sector [38]= new Kletka(5,4 ,1,1,0,0);
sector [39]= new Kletka(6,4 ,0,0,0,0);
sector [40]= new Kletka(7,4 ,0,1,0,1);
sector [41]= new Kletka(0,5 ,0,0,1,1);
sector [42]= new Kletka(1,5 ,1,1,1,0);
sector [43]= new Kletka(2,5 ,1,0,1,1);
sector [44]= new Kletka(3,5 ,1,1,1,0);
sector [45]= new Kletka(4,5 ,1,0,1,1);
sector [46]= new Kletka(5,5 ,1,1,1,1);
sector [47]= new Kletka(6,5 ,1,0,0,0);
sector [48]= new Kletka(7,5 ,0,1,0,1);
sector [49]= new Kletka(0,6 ,0,0,0,0);
sector [50]= new Kletka(1,6 ,0,1,0,1);
sector [51]= new Kletka(2,6 ,0,0,0,0);
sector [52]= new Kletka(3,6 ,0,1,0,1);
sector [53]= new Kletka(4,6 ,0,0,0,0);
sector [54]= new Kletka(5,6 ,0,1,0,1);
sector [55]= new Kletka(6,6 ,0,0,0,0);
sector [56]= new Kletka(7,6 ,0,1,0,1);
sector [57]= new Kletka(0,7 ,0,0,1,0);
sector [58]= new Kletka(1,7 ,1,0,0,1);
sector [59]= new Kletka(2,7 ,0,0,1,0);
sector [60]= new Kletka(3,7 ,1,0,1,1);
sector [61]= new Kletka(4,7 ,1,0,0,0);
sector [62]= new Kletka(5,7 ,0,0,1,1);
sector [63]= new Kletka(6,7 ,1,0,1,0);
sector [64]= new Kletka(7,7 ,1,0,0,1);
```

```

var klet = new Array();
var pos = new Array(64);
var put = new Array(64);
var napr = new Array(64);
var fir =1;
var last =1;
for(var i = 1;i<65;i++)
{
    pos[i]=0;
};
for(var i = 1;i<65;i++)
{
    put[i]="";
};
for(var i = 1;i<65;i++)
{
    napr[i]=0;
}
// var Text = script.readAll("input.txt");
// p1 = Text[0].indexOf(" ");
// p2 = Text[0].indexOf(" ",p1+1);
// p3 = Text[1].indexOf(" ");
// c1 = +Text[0].substring(0,p1);
// c2 = +Text[0].substring(p1+1,p2);
// c3 = +Text[0].substring(p2+1);
// c4 = +Text[1].substring(0,p3);
// c5 = +Text[1].substring(p3+1);
otv=Shir(6,1,3,2,4);

print(otv);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
var chik=0;
while (otv.length > 0)
{
    if (brick.sensor(A2).read()<25){chik++}
    sim = otv.substring(0,1);
    otv = otv.substring(1);
    if (sim=="F")
    {
        run(485);
    }
    if (sim=="L")
    {
        rotate(-81)
    }
    if (sim=="R")
    {
        rotate(85)
    }
}
brick.display().addLabel("finish",2,2);
brick.display().addLabel(chik,2,70);
brick.display().redraw();
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
script.wait(9999);

```



```

function Kletka(x,y,left,up,right,down)
{
    this.x=x;
    this.y=y;
    this.left=left;
    this.up=up;
    this.right=right;
    this.down=down;
}
function Shir(y1,x1,V,y2,x2)
{
    var S= x1+1 + y1*8;
    var F= x2+1 + y2*8;
    napr[S]=V;
    add(S);
    while (true) {
        ost = take();
        if (ost == F) {
            break;
        };
        if (sector[ost].left == 1) {
            if (napr[ost] == 0) {
                K = put[ost]+"F";
            };
            if (napr[ost] == 1) {
                K = put[ost]+"L"+"F";
            };
            if (napr[ost] == 2) {
                K = put[ost]+"LL"+"F";
            };
            if (napr[ost] == 3) {
                K = put[ost]+"R"+"F";
            };
            if (pos[ost-1] == 0) {
                put[ost-1] = K;
                add (ost-1);
                napr[ost-1] = 0;
            } else {
                if (K < put[ost-1]) {
                    put[ost-1] = K;
                    add (ost-1);
                    napr[ost-1] = 0;
                }
            }
        };
        if (sector[ost].up == 1) {
            if (napr[ost] == 0) {
                K = put[ost]+"R"+"F";
            };
            if (napr[ost] == 1) {
                K = put[ost]+"F";
            };
            if (napr[ost] == 2) {
                K = put[ost]+"L"+"F";
            };
            if (napr[ost] == 3) {
                K = put[ost]+"LL"+"F";
            };
        };
    }
}

```

```

        if (pos[ost+8] == 0) {
            put[ost+8] = K;
            add (ost+8);
            napr[ost+8] = 1;
        } else {
            if (K < put[ost+8]) {
                put[ost+8] = K;
                add (ost+8);
                napr[ost+8] = 1;
            }
        }
    };
if (sector[ost].right == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"LL"+"F";
    };
    if (napr[ost] == 1) {
        K = put[ost]+"R"+"F";
    };
    if (napr[ost] == 2) {
        K = put[ost]+"F";
    };
    if (napr[ost] == 3) {
        K = put[ost]+"L"+"F";
    };
    if (pos[ost+1] == 0) {
        put[ost+1] = K;
        add (ost+1);
        napr[ost+1] = 2;
    } else {
        if (K < put[ost+1]) {
            put[ost+1] = K;
            add (ost+1);
            napr[ost+1] = 2;
        }
    }
}
};
if (sector[ost].down == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"L"+"F";
    }
    if (napr[ost] == 1) {
        K = put[ost]+"LL"+"F";
    }
    if (napr[ost] == 2) {
        K = put[ost]+"R"+"F";
    }
    if (napr[ost] == 3) {
        K = put[ost]+"F";
    }
    if (pos[ost-8] == 0) {
        put[ost-8] = K;
        add (ost-8);
        napr[ost-8] = 3;
    } else {
        if (K < put[ost-8]) {
            put[ost-8] = K;
            add (ost-8);
            napr[ost-8] = 3;
        }
    }
}
}

```

```

        }
    }
    return put[F];
}
function add(q) {
    klet[last]=q;
    last=last+1;
    pos[q]=1;
};

function length() {
    return last-fir;
};

function take() {
    fir=fir+1;
    return klet[fir-1];
}
var run = function(length) {
    var leng = length;
    var p =60;
    var ob = 75 * Math.PI;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили энкодеры

    mr=brick.motor(M4);
    ml=brick.motor(M3);
    mr.setPower(p);
    ml.setPower(p);//запустили моторы

    var limit = (374/ob)*leng;
    var enc = 0;
    var k = 2;
    var razn = 0;
    while (enc < limit) {
        enc=(Math.abs(el.readRawData()) +
Math.abs(er.readRawData())) / 2;

        razn = (Math.abs(el.readRawData()) -
Math.abs(er.readRawData()))*k
        mr.setPower(p+razn);
        ml.setPower(p-razn);
        script.wait(5);
    }
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    script.wait(500);
}
var rotate = function(ar) {
    //run(20);
    var arc = ar;
    var k = 117;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили энкодеры
    mr=brick.motor(M4);
    ml=brick.motor(M3);

```

```

var ob = 75 * Math.PI;
if (arc < 0) {
    var leng = 2*k*Math.PI* -arc / 360;
    mr.setPower(60);
    ml.setPower(-60);
}
else {
    var leng = 2*k*Math.PI* arc / 360;
    mr.setPower(-60);
    ml.setPower(60);
}
var limit = (360/ob)*leng;
var enc = 0;
while (enc < limit) {
    enc=Math.abs(er.readRawData());
    script.wait(5);
}
mr.powerOff();
ml.powerOff();
script.wait(500);
}

var sector = new Array(64);
sector [1]= new Kletka(0,0 ,0,1,1,0);
sector [2]= new Kletka(1,0 ,1,0,1,0);
sector [3]= new Kletka(2,0 ,1,1,0,0);
sector [4]= new Kletka(3,0 ,0,0,1,0);
sector [5]= new Kletka(4,0 ,1,1,1,0);
sector [6]= new Kletka(5,0 ,1,0,0,0);
sector [7]= new Kletka(6,0 ,0,1,1,0);
sector [8]= new Kletka(7,0 ,1,0,0,0);
sector [9]= new Kletka(0,1 ,0,1,0,1);
sector [10]= new Kletka(1,1 ,0,0,0,0);
sector [11]= new Kletka(2,1 ,0,1,0,1);
sector [12]= new Kletka(3,1 ,0,0,0,0);
sector [13]= new Kletka(4,1 ,0,1,0,1);
sector [14]= new Kletka(5,1 ,0,0,0,0);
sector [15]= new Kletka(6,1 ,0,1,0,1);
sector [16]= new Kletka(7,1 ,0,0,0,0);
sector [17]= new Kletka(0,2 ,0,1,1,1);
sector [18]= new Kletka(1,2 ,1,0,1,0);
sector [19]= new Kletka(2,2 ,1,0,1,1);
sector [20]= new Kletka(3,2 ,1,0,1,0);
sector [21]= new Kletka(4,2 ,1,1,1,1);
sector [22]= new Kletka(5,2 ,1,0,1,0);
sector [23]= new Kletka(6,2 ,1,1,1,1);
sector [24]= new Kletka(7,2 ,1,1,0,0);
sector [25]= new Kletka(0,3 ,0,1,0,1);
sector [26]= new Kletka(1,3 ,0,0,1,0);
sector [27]= new Kletka(2,3 ,1,1,1,0);
sector [28]= new Kletka(3,3 ,1,0,0,0);
sector [29]= new Kletka(4,3 ,0,1,0,1);
sector [30]= new Kletka(5,3 ,0,0,0,0);
sector [31]= new Kletka(6,3 ,0,0,1,1);
sector [32]= new Kletka(7,3 ,1,1,0,1);
sector [33]= new Kletka(0,4 ,0,1,0,1);
sector [34]= new Kletka(1,4 ,0,0,0,0);
sector [35]= new Kletka(2,4 ,0,1,0,1);
sector [36]= new Kletka(3,4 ,0,0,0,0);
sector [37]= new Kletka(4,4 ,0,1,1,1);
sector [38]= new Kletka(5,4 ,1,1,0,0);
sector [39]= new Kletka(6,4 ,0,0,0,0);

```

```

sector [40]= new Kletka(7,4 ,0,1,0,1);
sector [41]= new Kletka(0,5 ,0,0,1,1);
sector [42]= new Kletka(1,5 ,1,1,1,0);
sector [43]= new Kletka(2,5 ,1,0,1,1);
sector [44]= new Kletka(3,5 ,1,1,1,0);
sector [45]= new Kletka(4,5 ,1,0,1,1);
sector [46]= new Kletka(5,5 ,1,1,1,1);
sector [47]= new Kletka(6,5 ,1,0,0,0);
sector [48]= new Kletka(7,5 ,0,1,0,1);
sector [49]= new Kletka(0,6 ,0,0,0,0);
sector [50]= new Kletka(1,6 ,0,1,0,1);
sector [51]= new Kletka(2,6 ,0,0,0,0);
sector [52]= new Kletka(3,6 ,0,1,0,1);
sector [53]= new Kletka(4,6 ,0,0,0,0);
sector [54]= new Kletka(5,6 ,0,1,0,1);
sector [55]= new Kletka(6,6 ,0,0,0,0);
sector [56]= new Kletka(7,6 ,0,1,0,1);
sector [57]= new Kletka(0,7 ,0,0,1,0);
sector [58]= new Kletka(1,7 ,1,0,0,1);
sector [59]= new Kletka(2,7 ,0,0,1,0);
sector [60]= new Kletka(3,7 ,1,0,1,1);
sector [61]= new Kletka(4,7 ,1,0,0,0);
sector [62]= new Kletka(5,7 ,0,0,1,1);
sector [63]= new Kletka(6,7 ,1,0,1,0);
sector [64]= new Kletka(7,7 ,1,0,0,1);

var klet = new Array();
var pos  = new Array(64);
var put  = new Array(64);
var napr = new Array(64);
var fir =1;
var last =1;
for(var i = 1;i<65;i++)
{
    pos[i]=0;
};
for(var i = 1;i<65;i++)
{
    put[i]="";
};
for(var i = 1;i<65;i++)
{
    napr[i]=0;
}
// var Text = script.readAll("input.txt");
// p1 = Text[0].indexOf(" ");
// p2 = Text[0].indexOf(" ",p1+1);
// p3 = Text[1].indexOf(" ");
// c1 = +Text[0].substring(0,p1);
// c2 = +Text[0].substring(p1+1,p2);
// c3 = +Text[0].substring(p2+1);
// c4 = +Text[1].substring(0,p3);
// c5 = +Text[1].substring(p3+1);
otv=Shir(1,2,1,4,2);

print(otv);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
var chik=0;

```

```

while (otv.length > 0)
{
    if (brick.sensor(A2).read()<26){chik++}
    sim = otv.substring(0,1);
    otv = otv.substring(1);
    if (sim=="F")
    {
        run(486);
    }
    if (sim=="L")
    {
        rotate(-80)
    }
    if (sim=="R")
    {
        rotate(79)
    }
}

brick.motor(M3).setPower(-40);

brick.motor(M4).setPower(-40);

script.wait(300);

brick.motor(M3).powerOff();

brick.motor(M4).powerOff();

if (brick.sensor(A2).read()<24){chik++}
brick.display().addLabel("finish",2,2);
brick.display().addLabel(chik,2,70);
brick.display().redraw();
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
script.wait(9999);

```

Программа команды для решения задачи второго этапа.

Решение задачи в Симуляторе: - решения нет

Решение задачи на реальном роботе:

```

function RRGGBB(R,G,B)
{
    this.R=R
    this.G=G;
    this.B=B;
}
function point(x,y)
{
    this.x=x;
    this.y=y;
}

```

```

function round_(r)
{
    return (Math.round(r*1)/1);
}

var n = 120;
var m = 160;
var photo = getPhoto();
var a = Array (120);

var k=0
for (i=0;i<120;i++)
{
    a[i] = Array (160);
    for (j=0;j<160;j++)
    {
        var R=((photo[k] & 0xff0000)>>16).toString();
        var G=((photo[k] & 0xff00)>>8).toString();
        var B=((photo[k] & 0xff).toString());
        summc = (R+G+B)/3;
        a[i,j] = summc;
        if (a[i,j]<=1000)
        {
            a[i,j]=0;
        }
        if (a[i,j]>1000)
        {
            a[i,j]=255
        }
        k++;
    }
}

var p1 = 0;
var p2 = 0;
var p3 = 0;
var p4 =0;

for (i=0;i<120;i++)
{
    for (j=0;j<160;j++)
    {
        if ((a[i,j]=0)&&(p1==0)) {p1=i}
        if (a[i,j]=0){p2=i}
    }
}

for (j=0;j<160;j++)
{

```

```

        for (i=0;i<120;i++)
        {
                if ((a[i,j]=0)&&(p3==0)) {p3=j}
                if (a[i,j]=0){p4=j}
        }
}
var x= round_(p1+p2)/2;
var y= round_(p4+p3)/2;
var raznX =round_(Math.abs(p1-p2)/5);
var raznY =round_(Math.abs(p3-p4)/5);
var summ=0;
if (a[x+raznX,y+raznY])
{
        if (a[x,y-raznY]=1){summ+=8}
        if (a[x-raznX,y]=1){summ+=4}
        if (a[x+raznX,y]=1){summ+=2}
        if (a[x,y+raznY]=1){summ+=1}
}
if (a[x+raznX,y-raznY])
{
        if (a[x-raznX,y]=1){summ+=8}
        if (a[x,y+raznY]=1){summ+=4}
        if (a[x,y-raznY]=1){summ+=2}
        if (a[x+raznX,y]=1){summ+=1}
}
if (a[x-raznX,y+raznY])
{
        if (a[x+raznX,y]=1){summ+=8}
        if (a[x,y-raznY]=1){summ+=4}
        if (a[x,y+raznY]=1){summ+=2}
        if (a[x-raznX,y]=1){summ+=1}
}

```



```

    }
    if (a[x-raznX,y-raznY])
    {
        if (a[x,y+raznY]=1) {summ+=8}
        if (a[x+raznX,y]=1) {summ+=4}
        if (a[x-raznX,y]=1) {summ+=2}
        if (a[x,y-raznY]=1) {summ+=1}
    }
    brick.display().addLabel(summ,70,200);
    brick.display().redraw();
    script.wait(2000);
    for (i=0;i<120;i++)
    {
        for (j=0;j<160;j++)
        {
            if (a[i,j]<=100)
            {
                brick.display().setPainterColor("black");
                brick.display().drawPoint(i,j);
                brick.display().redraw();
            }
            if (a[i,j]>100)
            {
                brick.display().setPainterColor("white");
                brick.display().drawPoint(i,j);
                brick.display().redraw();
            }
        }
    }
}

```

```
    }  
    script.wait(10000);
```

```
function RRGGBB(R,G,B)
```

```
{  
  
    this.R=R  
    this.G=G;  
    this.B=B;  
}
```

```
function point(x,y)
```

```
{  
    this.x=x;  
    this.y=y;  
}
```

```
function curcol(x,y,r)
```

```
{  
    for (i = x - r ; i < x + r ; i++ )  
    {  
        for (j = y - r ; j < y + r ; j++ )  
        {  
            if ((i*i+j*j)=r*r)  
            {  
  
            }  
        }  
    }  
}
```

```
function round_(r)
```

```
{
```

```

return (Math.round(r*1)/1);
}

var n = 120;
var m = 160;
var list = script.readAll("input.txt");
var s = list[0];
var photo = list[1];
var photol = list[2];
var a = Array (120);
for (i=0;i<120;i++)
{
    a[i] = Array (160);
    for (j=0;j<160;j++)
    {
        var st=0;
        var fin=photo.indexOf(" ");
        var rgb = +photo.substring(st,fin);
        var R=((rgb & 0xff0000)>>16).toString();
        var G=((rgb & 0xff00)>>8).toString();
        var B=((rgb & 0xff).toString());
        summc = (R+G+B)/3;
        a[i,j] = summc;
        photo = photo.substring(fin+1);
        if (a[i,j]<=1000)
        {
            a[i,j]=0;
        }
        if (a[i,j]>1000)
        {
            a[i,j]=255

```

```

        }

    }

var p1 = 0;
var p2 = 0;
var p3 = 0;
var p4 =0;
for (i=0;i<120;i++)
{
    for (j=0;j<160;j++)
    {
        if ((a[i,j]=0)&&(p1==0)) {p1=i}
        if (a[i,j]=0){p2=i}
    }
}
for (j=0;j<160;j++)
{
    for (i=0;i<120;i++)
    {
        if ((a[i,j]=0)&&(p3==0)) {p3=j}
        if (a[i,j]=0){p4=j}
    }
}
var x= round_(p1+p2)/2;
var y= round_(p4+p3)/2;
var raznX =round_(Math.abs(p1-p2));
var raznY =round_(Math.abs(p3-p4));
var summ=0;
if a[x+raznX,y+raznY]
{

```

```
        if (a[x,y-raznY]=1) {summ+=8}
        if (a[x-raznX,y]=1) {summ+=4}
        if (a[x+raznX,y]=1) {summ+=2}
        if (a[x,y+raznY]=1) {summ+=1}
    }
    if a[x+raznX,y-raznY]
    {
        if (a[x-raznX,y]=1) {summ+=8}
        if (a[x,y+raznY]=1) {summ+=4}
        if (a[x,y-raznY]=1) {summ+=2}
        if (a[x+raznX,y]=1) {summ+=1}
    }
    if a[x-raznX,y+raznY]
    {
        if (a[x+raznX,y]=1) {summ+=8}
        if (a[x,y-raznY]=1) {summ+=4}
        if (a[x,y+raznY]=1) {summ+=2}
        if (a[x-raznX,y]=1) {summ+=1}
    }
    if a[x-raznX,y-raznY]
    {
        if (a[x,y+raznY]=1) {summ+=8}
        if (a[x+raznX,y]=1) {summ+=4}
        if (a[x-raznX,y]=1) {summ+=2}
        if (a[x,y-raznY]=1) {summ+=1}
    }
    brick.display().addLabel(sum,10,10);
    brick.display().redraw();
```

Программа команды для решения задачи третьего этапа.

Решение задачи в Симуляторе:

```
function foto(TXT) {

var file = TXT;

    var photo = [];

    // перевести бинарное число в десятичное

    function count(num1, num2, num3, num4) { return (num1 *
Math.pow(2,3) + num2 * Math.pow(2,2) + num3 * Math.pow(2,1) + num4 *
Math.pow(2,0)); }

    // функция берёт сырое изображение и на выходе даёт в переменной
photo бинарное обрезанное изображение

    function reader_checker(raw_photo){

        n = 120;

        m = 159;

        var z = 0;

        var x_min, x_max, y_max, y_min;

        for (var i = 0; i < n; i++){

            photo[i] = [];

            z++;

            // бинаризация изображения

            for (var j = 0; j < m; j++ ){

                z++

                var temp = (((raw_photo[z] & 0xff0000) >> 18)
+ ((raw_photo[z] & 0xff00) >> 10) + ((raw_photo[z] & 0xff) >> 2));

                photo[i].push(temp < 40 ? 1 : 0);

            }

        }

    }

}
```

```

        }
    }

    m = 155;
    var i = 0;
    var j = 0;
    while((i < n)){
        if (photo[i][j] == 1){
            xmin = i;
            break;
        }
        i += (++j/m == 1) ? 1: 0;
        j %= m;
    }

    i = n - 1;
    j = 0;
    while((i >= 0)){
        if (photo[i][j] == 1){
            xmax = i;
            break;
        }
        i -= (++j/m == 1) ? 1: 0;
        j %= m;
    }

    i = 0;
    j = 0;
    while((i < m)){

```

```
        if (photo[j][i] == 1){
            ymin = i;
            break;
        }
        i += (++j/n == 1) ? 1: 0;
        j %= n;
    }
```

```
i = m - 1;
j = 0;
while((i >= 0)){
    if (photo[j][i] == 1){
        ymax = i;
        break;
    }
    i -= (++j/n == 1) ? 1: 0;
    j %= n;
}
```

```
var x = 0;
var y = 0;
var temp_photo = [];
for (i = xmin; i <= xmax; i++){
    temp_photo[x] = [];
    for (var j = ymin; j <= ymax; j++){
        temp_photo[x][y] = photo[i][j];
        y++;
    }
}
```



```

    }

    //print(temp_photo[x]);

    x++;

    y = 0;

}

photo = temp_photo;

// новые размеры изображения
n = xmax - xmin + 1;
m = ymax - ymin + 1;

// если не нашли маркер
return ((n<2)|| (m<2)) ? -1 : 1;
}

// находим средню точку между двумя точками
function find_middle_point(t_x1, t_y1, t_x2, t_y2){
    var t_output = [];

    t_output.push(Math.floor((t_x2 - t_x1) * 0.5 + t_x1));
    t_output.push(Math.floor((t_y2 - t_y1) * 0.5 + t_y1));

    return t_output;
}

function count_center(x1, x2, x3, x4, y1, y2, y3, y4) {
    var x, y;

    var a1 = y1 - y3;
    var b1 = x3 - x1;

    var c1 = - (a1 * x1 + b1 * y1);

    var a2 = y2 - y4;
    var b2 = x4 - x2;

```

```

var c2 = -(a2 * x2 + b2 * y2);
x = -(c1 * b2 - c2 * b1) / (a1 * b2 - a2 * b1);
y = -(a1 * c2 - a2 * c1) / (a1 * b2 - a2 * b1);
var t_output = [];
t_output.push(Math.floor(x));
t_output.push(Math.floor(y));
return t_output;
}

```

```

function detect() {
    var x1, x2, x3, x4, y1, y2, y3, y4;

    var i = 0;
    var j = 0;
    var k = 0;
    while(i < n){
        if (photo[k][j] == 1){
            x1 = k;
            y1 = j;
            break;
        }
        i += (--k < 0) ? 1 : 0;
        k = (k < 0) ? i : k;
        j++;
        j %= m;
    }

    i = m - 1;
    j = 0;
    k = m - 1;

```

```

while(i > 0){
    if (photo[j][k] == 1){
        x2 = j;
        y2 = k;
        break;
    }
    i -= (++k >= m) ? 1: 0;
    k = (k >= m) ? i : k;
    j++;
    j %= n;
}

```

```

i = n - 1;
k = n - 1;
j = m - 1;
while( i >= 0){
    if (photo[k][j] == 1){
        x3 = k;
        y3 = j;
        break;
    }

    i -= (++k >= n) ? 1: 0;
    k = (k >= n) ? i : k;
    j--;
    j = (j <0) ? m-1 : j;
}

```

```
i = 0;
k = 0;
j = n - 1;
while( i < m){
    if (photo[j][k] == 1){
        x4 = j;
        y4 = k;
        break;
    }

    i += (--k < 0) ? 1: 0;
    k = (k < 0) ? i : k;
    j--;
    j = (j < 0) ? n-1 : j;
}
```

```
var center_x, center_y;
var res = [];
res = count_center(x1, x2, x3, x4, y1, y2, y3, y4);
center_x = res[0];
center_y = res[1];
var x_top = Math.floor((x1 + x2) / 2);
var y_top = Math.floor((y1 + y2) / 2);
var x_left = Math.floor((x1 + x4) / 2);
var y_left = Math.floor((y1 + y4) / 2);
var x_down = Math.floor((x3 + x4) / 2);
var y_down = Math.floor((y3 + y4) / 2);
var x_right = Math.floor((x2 + x3) / 2);
var y_right = Math.floor((y2 + y3) / 2);
```

```

var col = [];

col[0] = [];

col[0] = find_middle_point(center_x, center_y, x1, y1);

col[1] = [];

col[1] = find_middle_point(center_x, center_y, x2, y2);

col[2] = [];

col[2] = find_middle_point(center_x, center_y, x3, y3);

col[3] = [];

col[3] = find_middle_point(center_x, center_y, x4, y4);

var temp_number1 = find_middle_point(center_x, center_y,
x_top, y_top);

var temp_number2 = find_middle_point(center_x, center_y,
x_left, y_left);

var temp_number3 = find_middle_point(center_x, center_y,
x_right, y_right);

var temp_number4 = find_middle_point(center_x, center_y,
x_down, y_down);

if (photo[col[0][0]][col[0][1]] == 0)

    return
count(photo[temp_number4[0]][temp_number4[1]],
photo[temp_number3[0]][temp_number3[1]],
photo[temp_number2[0]][temp_number2[1]],
photo[temp_number1[0]][temp_number1[1]]);

else if (photo[col[1][0]][col[1][1]] == 0)

    return
count(photo[temp_number2[0]][temp_number2[1]],
photo[temp_number4[0]][temp_number4[1]],
photo[temp_number1[0]][temp_number1[1]],
photo[temp_number3[0]][temp_number3[1]]);

else if (photo[col[2][0]][col[2][1]] == 0)

    return
count(photo[temp_number1[0]][temp_number1[1]],
photo[temp_number2[0]][temp_number2[1]],
photo[temp_number3[0]][temp_number3[1]],
photo[temp_number4[0]][temp_number4[1]]);

else

```

```

        return
count(photo[temp_number3[0]][temp_number3[1]],
photo[temp_number1[0]][temp_number1[1]],
photo[temp_number4[0]][temp_number4[1]],
photo[temp_number2[0]][temp_number2[1]]);

    }

    /*

function cameraDetect(_raw_photo){

    var error=reader_checker(_raw_photo);

    if (error>0)

    {counter = detect();

    brick.display().addLabel(counter,10,10);

    brick.display().redraw();

    print(counter);

    return counter

    script.wait(5000);

    }

}*/

var f_x_end = -1;

var f_y_end = -1

var temp_coord = reader_checker(file[1].split(" ")) > 0 ?
detect(): -1;

if (temp_coord > 7){

    f_x_end = reader_checker(file[2].split(" ")) > 0 ?
detect(): -1;

    f_y_end = temp_coord - 8;

}else{

    f_x_end = temp_coord;

    f_y_end = reader_checker(file[2].split(" ")) > 0 ?
(detect() - 8): -1 ;

}

```

```

XEND=f_x_end;

YEND=f_y_end;

print(f_x_end + " " + f_y_end);

}
//
=====
=====
//
=====
=====
//
=====
=====
=====

```

```

function Kletka(x,y,left,up,right,down,mark)
{
    this.x=x;
    this.y=y;
    this.left=left;
    this.up=up;
    this.right=right;
    this.down=down;
    this.mark=mark;
}
function Shir(y1,x1,V,y2,x2)
{
    fir =1;
    last =1;
    for(var i = 1;i<65;i++)
    {
        pos[i]=0;
    };
    for(var i = 1;i<65;i++)
    {
        put[i]="";
    };
    for(var i = 1;i<65;i++)
    {
        napr[i]=0;
    }
    var S= x1+1 + y1*8;
    var F= x2+1 + y2*8;
    napr[S]=V;
    add(S);
    while (true) {
        ost = take();
        if (ost == F) {
            break;
        };
        if (sector[ost].left == 1) {
            if (napr[ost] == 0) {
                K = put[ost]+"F";
            };
            if (napr[ost] == 1) {
                K = put[ost]+"L"+"F";
            };
        };
    };
}

```



```

//          } else {
//          if (K < put[ost+1]) {
//          put[ost+1] = K;
//          add (ost+1);
//          napr[ost+1] = 2;
//          }
//          }
};
if (sector[ost].down == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"L"+"F";
    }
    if (napr[ost] == 1) {
        K = put[ost]+"LL"+"F";
    }
    if (napr[ost] == 2) {
        K = put[ost]+"R"+"F";
    }
    if (napr[ost] == 3) {
        K = put[ost]+"F";
    }
    if (pos[ost-8] == 0) {
        put[ost-8] = K;
        add (ost-8);
        napr[ost-8] = 3;
//          } else {
//          if (K < put[ost-8]) {
//          put[ost-8] = K;
//          add (ost-8);
//          napr[ost-8] =3;
//          }
//          }
    }
    if (length() == 0) {
        return 0;
    }
}
return put[F];
}

function add(q) {
    klet[last]=q;
    last=last+1;
    pos[q]=1;
};

function length() {
    return last-fir;
};

function take() {
    fir=fir+1;
    return klet[fir-1];
}
var run = function(length) {
    var leng = length;
    var p =90;
    var ob = 75 * Math.PI;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);

```

```

el.reset();
er.reset();// обнулили энкодеры

mr=brick.motor(M4);
ml=brick.motor(M3);
mr.setPower(p);
ml.setPower(p);//запустили моторы

var limit = (374/ob)*leng;
var enc = 0;
var k = 1;
var D = 1;
//
var df = 0.4;
var l = 4;
var razn = 0;
var razn_old = 0;
while (enc < limit) {
    if (brick.sensor(A2).read() < 29) { //езда по правой руке
        razn = (brick.sensor(A2).read()-28);
        g = razn*l + (razn-razn_old)*df;
        mr.setPower(p-g);
        ml.setPower(p+g);
        razn_old=razn;
    } else {
        if (brick.sensor(A3).read() < 29) { //езда по левой
руке

                razn = (brick.sensor(A3).read()-28);
                g = razn*l + (razn-razn_old)*df;
                mr.setPower(p+g);
                ml.setPower(p-g);
                razn_old=razn;
            }
        }
    if ((brick.sensor(A2).read() >
26)&&(brick.sensor(A3).read() > 26)) {//езда вперед

            razn = (Math.abs(el.readRawData())-
Math.abs(er.readRawData()))
            g = razn*k + (razn-razn_old)*D;
            mr.setPower(p+g);
            ml.setPower(p-g);
            razn_old=razn;
        }
        enc=(Math.abs(el.readRawData()) +
Math.abs(er.readRawData())) / 2;
        script.wait(10);
    }
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    script.wait(500);
}

function deshifr(str) {
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    while (str.length > 0)
    {
        if (tNap > 0) {

```

```

        tNap= tNap % 4
    }
    if (tNap < 0) {
        tNap = 4 - Math.abs(tNap % 4)
    }
    sim = str.substring(0,1);
    str = str.substring(1);
    ERR=0;

    print(brick.sensor(A1).read());
    if (brick.sensor(A2).read()> 55) {
        if (tNap == 0) {
            if (sector[tX*8+1+tY].down == 1) {
                if (sector[tX*8+1+tY-8].mark == 0) {
                    sector[tX*8+1+tY-8].mark = 1;
                    ERR = 1;
                    print("mL")}
            }
        }
        if (tNap== 1) {
            if (sector[tX*8+1+tY].left == 1) {
                if (sector[tX*8+1+tY-1].mark == 0) {
                    sector[tX*8+1+tY-1].mark = 1;
                    ERR = 1;
                    print("mL")}
            }
        }
        if (tNap== 2) {
            if (sector[tX*8+1+tY].up == 1) {
                if (sector[tX*8+1+tY+8].mark == 0) {
                    sector[tX*8+1+tY+8].mark = 1;
                    ERR = 1;
                    print("mL")}
            }
        }
        if (tNap== 3) {
            if (sector[tX*8+1+tY].right == 1) {
                if (sector[tX*8+1+tY+1].mark == 0) {
                    sector[tX*8+1+tY+1].mark = 1;
                    ERR = 1;
                    print("mL")}
            }
        }
    }
};

    if (brick.sensor(A3).read()> 55) {
        if (tNap == 0) {
            if (sector[tX*8+1+tY].up == 1) {
                if (sector[tX*8+1+tY+8].mark == 0) {
                    sector[tX*8+1+tY+8].mark = 1;
                    ERR = 1;
                    print("mR")}
            }
        }
        if (tNap== 1) {
            if (sector[tX*8+1+tY].right == 1) {
                if (sector[tX*8+1+tY+1].mark == 0) {
                    sector[tX*8+1+tY+1].mark = 1;
                    ERR = 1;
                    print("mR")}
            }
        }
    }
};

```

```

}
if (tNap== 2) {
    if (sector[tX*8+1+tY].down == 1) {
        if (sector[tX*8+1+tY-8].mark == 0) {
            sector[tX*8+1+tY-8].mark = 1;
            ERR = 1;
            print("mR")}
        }
    }
}
if (tNap== 3) {
    if (sector[tX*8+1+tY].left == 1) {
        if (sector[tX*8+1+tY-1].mark == 0) {
            sector[tX*8+1+tY-1].mark = 1;
            ERR = 1;
            print("mR")}
        }
    }
}
};

if (brick.sensor(A1).read() < 55) {
    if (tNap == 0) {
        if (sector[tX*8+1+tY].left == 1) {

            sector[tX*8+1+tY].left = 0;
            ERR = 1;
            print("erF")
        }
    }
    if (tNap== 1) {
        if (sector[tX*8+1+tY].up == 1) {
            sector[tX*8+1+tY].up = 0;
            ERR = 1;
            print("erF")
        }
    }
    if (tNap== 2) {
        if (sector[tX*8+1+tY].right == 1) {
            sector[tX*8+1+tY].right = 0;
            ERR = 1;
            print("erF")
        }
    }
    if (tNap== 3) {
        if (sector[tX*8+1+tY].down == 1) {
            sector[tX*8+1+tY].down = 0;
            ERR = 1;
            print("erF")
        }
    }
}
};

if (ERR == 1) {
    break;
}

if (sim=="F")
{
    run(890);
    if (tNap == 0) {tY=tY-1};
    if (tNap == 1) {tX=tX+1};
    if (tNap == 2) {tY=tY+1};
    if (tNap == 3) {tX=tX-1};
}

```

```

    }
    if (sim=="L")
    {
        if (brick.sensor(A1).read()<25) {
            brick.motor(M4).setPower(-60);
            brick.motor(M3).setPower(-60);
            script.wait(200);
            brick.motor(M4).powerOff();
            brick.motor(M3).powerOff();
        }
        rotate(88)
        tNap=tNap-1;
    }
    if (sim=="R")
    {
        if (brick.sensor(A1).read()<25) {
            brick.motor(M4).setPower(-60);
            brick.motor(M3).setPower(-60);
            script.wait(200);
            brick.motor(M4).powerOff();
            brick.motor(M3).powerOff();
        }
        tNap=tNap+1;
        rotate(-88)
    }
}

}

var rotate = function(ar) {
    //run(20);
    var arc = ar;
    var k = 117;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили енкодеры
    mr=brick.motor(M4);
    ml=brick.motor(M3);
    var ob = 75 * Math.PI;
    if (arc < 0) {
        var leng = 2*k*Math.PI* -arc / 360;
        mr.setPower(60);
        ml.setPower(-60);
    }
    else {
        var leng = 2*k*Math.PI* arc / 360;
        mr.setPower(-60);
        ml.setPower(60);
    }
    var limit = (360/ob)*leng;
    var enc = 0;
    while (enc < limit) {
        enc=Math.abs(er.readRawData());
        script.wait(5);
    }
    mr.powerOff();
    ml.powerOff();
    script.wait(500);
}

var sector = new Array(64);
sector [1]= new Kletka(0,0 ,0,1,1,0, 0);

```

```
sector [2]= new Kletka(1,0 ,1,0,1,0, 0);
sector [3]= new Kletka(2,0 ,1,1,0,0, 0);
sector [4]= new Kletka(3,0 ,0,0,1,0, 0);
sector [5]= new Kletka(4,0 ,1,1,1,0, 0);
sector [6]= new Kletka(5,0 ,1,0,0,0, 0);
sector [7]= new Kletka(6,0 ,0,1,1,0, 0);
sector [8]= new Kletka(7,0 ,1,0,0,0, 0);
sector [9]= new Kletka(0,1 ,0,1,0,1, 0);
sector [10]= new Kletka(1,1 ,0,0,0,0, 0);
sector [11]= new Kletka(2,1 ,0,1,0,1, 0);
sector [12]= new Kletka(3,1 ,0,0,0,0, 0);
sector [13]= new Kletka(4,1 ,0,1,0,1, 0);
sector [14]= new Kletka(5,1 ,0,0,0,0, 0);
sector [15]= new Kletka(6,1 ,0,1,0,1, 0);
sector [16]= new Kletka(7,1 ,0,0,0,0, 0);
sector [17]= new Kletka(0,2 ,0,1,1,1, 0);
sector [18]= new Kletka(1,2 ,1,0,1,0, 0);
sector [19]= new Kletka(2,2 ,1,0,1,1, 0);
sector [20]= new Kletka(3,2 ,1,0,1,0, 0);
sector [21]= new Kletka(4,2 ,1,1,1,1, 0);
sector [22]= new Kletka(5,2 ,1,0,1,0, 0);
sector [23]= new Kletka(6,2 ,1,1,1,1, 0);
sector [24]= new Kletka(7,2 ,1,1,0,0, 0);
sector [25]= new Kletka(0,3 ,0,1,0,1, 0);
sector [26]= new Kletka(1,3 ,0,0,1,0, 0);
sector [27]= new Kletka(2,3 ,1,1,1,0, 0);
sector [28]= new Kletka(3,3 ,1,0,0,0, 0);
sector [29]= new Kletka(4,3 ,0,1,0,1, 0);
sector [30]= new Kletka(5,3 ,0,0,0,0, 0);
sector [31]= new Kletka(6,3 ,0,0,1,1, 0);
sector [32]= new Kletka(7,3 ,1,1,0,1, 0);
sector [33]= new Kletka(0,4 ,0,1,0,1, 0);
sector [34]= new Kletka(1,4 ,0,0,0,0, 0);
sector [35]= new Kletka(2,4 ,0,1,0,1, 0);
sector [36]= new Kletka(3,4 ,0,0,0,0, 0);
sector [37]= new Kletka(4,4 ,0,1,1,1, 0);
sector [38]= new Kletka(5,4 ,1,1,0,0, 0);
sector [39]= new Kletka(6,4 ,0,0,0,0, 0);
sector [40]= new Kletka(7,4 ,0,1,0,1, 0);
sector [41]= new Kletka(0,5 ,0,0,1,1, 0);
sector [42]= new Kletka(1,5 ,1,1,1,0, 0);
sector [43]= new Kletka(2,5 ,1,0,1,1, 0);
sector [44]= new Kletka(3,5 ,1,1,1,0, 0);
sector [45]= new Kletka(4,5 ,1,0,1,1, 0);
sector [46]= new Kletka(5,5 ,1,1,1,1, 0);
sector [47]= new Kletka(6,5 ,1,0,0,0, 0);
sector [48]= new Kletka(7,5 ,0,1,0,1, 0);
sector [49]= new Kletka(0,6 ,0,0,0,0, 0);
sector [50]= new Kletka(1,6 ,0,1,0,1, 0);
sector [51]= new Kletka(2,6 ,0,0,0,0, 0);
sector [52]= new Kletka(3,6 ,0,1,0,1, 0);
sector [53]= new Kletka(4,6 ,0,0,0,0, 0);
sector [54]= new Kletka(5,6 ,0,1,0,1, 0);
sector [55]= new Kletka(6,6 ,0,0,0,0, 0);
sector [56]= new Kletka(7,6 ,0,1,0,1, 0);
sector [57]= new Kletka(0,7 ,0,0,1,0, 0);
sector [58]= new Kletka(1,7 ,1,0,0,1, 0);
sector [59]= new Kletka(2,7 ,0,0,1,0, 0);
sector [60]= new Kletka(3,7 ,1,0,1,1, 0);
sector [61]= new Kletka(4,7 ,1,0,0,0, 0);
sector [62]= new Kletka(5,7 ,0,0,1,1, 0);
```

```

sector [63]= new Kletka(6,7 ,1,0,1,0, 0);
sector [64]= new Kletka(7,7 ,1,0,0,1, 0);

var klet = new Array();
var pos  = new Array(64);
var put  = new Array(64);
var napr = new Array(64);
var fir;
var last;

var XEND = 0;
var YEND = 0;
var TEXT = script.readAll("input.txt");
foto(TEXT);
// p1 = Line1.indexOf(" ");
// p2 = Line1.indexOf(" ",p1+1);
// p3 = Line1.indexOf(" ");
// c1 = Line1.substring(0,p1);
// c2 = Line1.substring(p1+1,p2);
// c3 = Line1.substring(p2+1);
// c4 = +Text[1].substring(0,p3);
// c5 = +Text[1].substring(p3+1);

//XX - x координата финиша; YY - y координата финиша; tNap - направление в
данный момент;
// tX - текущая координата x; tY -текущая координата y;
var chik;
var flag = true;//финишировал за 449 секунд; 453 секунды , примерно 7,55
минут; 370 секунд;
var Line1 = TEXT[0];
var Line2 = TEXT[1];
var Line3 = TEXT[2];
p1 = +Line1.indexOf(" ");
p2 = +Line1.indexOf(" ",p1+1);
p3 = +Line1.indexOf(" ");
c1 = +Line1.substring(0,p1);
c2 = +Line1.substring(p1+1,p2);
c3 = +Line1.substring(p2+1);
var tX= c1;
var tY= c2;
var tNap= c3;

var flag = true;//финишировал за 449 секунд; 453 секунды , примерно 7,55
минут; 370 секунд;
while (flag) {
    flag = false;
    mindlin="a";
    var tm;
    for (var i=0;i<8;i++) {
        for (var j=0;j<8;j++) {
            if ((sector[i*8+1+j].mark!=1) &&
(sector[i*8+1+j].mark!=2)) {
                tm = Shir(tX,tY,tNap,i,j);
                if (tm == 0) {
                    sector[i*8+1+j].mark=2;
                } else {
                    if (tm < mindlin) {
                        flag = true;
                        mindlin=tm;
                    }
                }
            }
        }
    }
}

```

```

script.wait(10);
}
}
}
}
// print(mindlin);
sector[tX*8+1+tY].mark=1;
deshifr(mindlin);
}
var zabl = 0;
for (var i=0;i<8;i++) {
    for (var j=0;j<8;j++) {
        if (sector[i*8+1+j].mark==2) {
            zabl +=1;
        }
    }
}
zabl = zabl - 12;
print(zabl);
print("stop");

deshifr(Shir(tX,tY,tNap,XEND,YEND));

brick.motor(M3).setPower(-40);
brick.motor(M4).setPower(-40);
script.wait(300);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
brick.display().addLabel(zabl,2,2);
brick.display().redraw();
script.wait(9999);

```

Решение задачи на реальном роботе:

```

function Kletka(x,y,left,up,right,down)
{
    this.x=x;
    this.y=y;
    this.left=left;
    this.up=up;
    this.right=right;
    this.down=down;
}
function Shir(y1,x1,V,y2,x2)
{
    fir =1;
    last =1;
    for(var i = 1;i<65;i++)
    {
        pos[i]=0;
    }
}

```



```

};

for(var i = 1;i<65;i++)

{

    put[i]="";

};

for(var i = 1;i<65;i++)

{

    napr[i]=0;

}

var S= x1+1 + y1*8;
var F= x2+1 + y2*8;
napr[S]=V;
add(S);
while (true) {
    ost = take();
    if (ost == F) {
        break;
    };
    if (sector[ost].left == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"F";
        };
        if (napr[ost] == 1) {
            K = put[ost]+"L"+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"LL"+"F";
        };
        if (napr[ost] == 3) {
            K = put[ost]+"R"+"F";
        };
        if (pos[ost-1] == 0) {
            put[ost-1] = K;
            add (ost-1);
            napr[ost-1] = 0;
        } else {
            if (K < put[ost-1]) {
                put[ost-1] = K;
                add (ost-1);
                napr[ost-1] = 0;
            }
        }
    };
    if (sector[ost].up == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"R"+"F";
        };
        if (napr[ost] == 1) {
            K = put[ost]+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"L"+"F";
        };
    };
};

```

```

        if (napr[ost] == 3) {
            K = put[ost]+"LL"+"F";
        };
        if (pos[ost+8] == 0) {
            put[ost+8] = K;
            add (ost+8);
            napr[ost+8] = 1;
        } else {
            if (K < put[ost+8]) {
                put[ost+8] = K;
                add (ost+8);
                napr[ost+8] = 1;
            }
        }
    };
    if (sector[ost].right == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"LL"+"F";
        };
        if (napr[ost] == 1) {
            K = put[ost]+"R"+"F";
        };
        if (napr[ost] == 2) {
            K = put[ost]+"F";
        };
        if (napr[ost] == 3) {
            K = put[ost]+"L"+"F";
        };
        if (pos[ost+1] == 0) {
            put[ost+1] = K;
            add (ost+1);
            napr[ost+1] = 2;
        } else {
            if (K < put[ost+1]) {
                put[ost+1] = K;
                add (ost+1);
                napr[ost+1] = 2;
            }
        }
    };
    if (sector[ost].down == 1) {
        if (napr[ost] == 0) {
            K = put[ost]+"L"+"F";
        }
        if (napr[ost] == 1) {
            K = put[ost]+"LL"+"F";
        }
        if (napr[ost] == 2) {
            K = put[ost]+"R"+"F";
        }
        if (napr[ost] == 3) {
            K = put[ost]+"F";
        }
        if (pos[ost-8] == 0) {
            put[ost-8] = K;
            add (ost-8);
            napr[ost-8] = 3;
        } else {
            if (K < put[ost-8]) {
                put[ost-8] = K;
                add (ost-8);
            }
        }
    }
}

```

```

        napr[ost-8] =3;
    }
}

    }
}
return put[F];
}

function add(q) {
    klet[last]=q;
    last=last+1;
    pos[q]=1;
};

function length() {
    return last-fir;
};

function take() {
    fir=fir+1;
    return klet[fir-1];
}
var run = function(length) {
    var leng = length;
    var p =60;
    var ob = 75 * Math.PI;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили енкодеры

    mr=brick.motor(M4);
    ml=brick.motor(M3);
    mr.setPower(p);
    ml.setPower(p);//запустили моторы

    var limit = (374/ob)*leng;
    var enc = 0;
    var k = 1;
    var D = 1;
    //
    var df = 0.4;
    var l = 4;
    var razn = 0;
    var razn_old = 0;
    while (enc < limit) {
        if (brick.sensor(A2).read() < 18) { //езда по правой руке
            razn = (brick.sensor(A2).read()-14);
            g = razn*l + (razn-razn_old)*df;
            mr.setPower(p-g);
            ml.setPower(p+g);
            razn_old=razn;
        } else {
            if (brick.sensor(A3).read() < 18) { //езда по левой
руке

                razn = (brick.sensor(A3).read()-14);
                g = razn*l + (razn-razn_old)*df;
                mr.setPower(p+g);
            }
        }
    }
}

```

```

        ml.setPower(p-g);
        razn_old=razn;
    }
    }
    if ((brick.sensor(A2).read() >
18)&&(brick.sensor(A3).read() > 18)) {//езда вперед

        razn = (Math.abs(el.readRawData()) -
Math.abs(er.readRawData()))
        g = razn*k + (razn-razn_old)*D;
        mr.setPower(p+g);
        ml.setPower(p-g);
        razn_old=razn;
    }
    enc=(Math.abs(el.readRawData()) +
Math.abs(er.readRawData())) / 2;
    script.wait(10);
    }
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    script.wait(500);
}
function proezd(x1,y1,N,x2,y2) {
    fir =1;
    last =1;
    for(var i = 1;i<65;i++)
    {
        pos[i]=0;
    };
    for(var i = 1;i<65;i++)
    {
        put[i]="";
    };
    for(var i = 1;i<65;i++)
    {
        napr[i]=0;
    }
    otv=Shir(x1,y1,N,x2,y2);
    tX = x1;
    tY = y1;
    XX = x2;
    YY = y2;
    tNap = N;
    print(otv);
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    while (otv.length > 0)
    {
        if (tNap > 0) {
            tNap= tNap % 4
        }
        if (tNap < 0) {
            tNap = 4 + (tNap % 4)
        }
    }

    if (brick.sensor(A2).read()<26){chik++}
    sim = otv.substring(0,1);
    otv = otv.substring(1);
}

```

```

if (sim=="F")
{
    run(515);
    if (tNap == 0) {tY=tY-1};
    if (tNap == 1) {tX=tX+1};
    if (tNap == 2) {tY=tY+1};
    if (tNap == 3) {tX=tX-1};

}
if (sim=="L")
{
    rotate(-79)
    tNap=tNap-1;
}
if (sim=="R")
{
    tNap=tNap+1;
    rotate(79)
}
print (tNap + " " + tX + " " + tY);
}

}

function deshifr(str) {
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
while (str.length > 0)
{
    if (tNap > 0) {
        tNap= tNap % 4
    }
    if (tNap < 0) {
        tNap = 4 - Math.abs(tNap % 4)
    }
sim = str.substring(0,1);
str = str.substring(1);
ERR=0;
if ((brick.sensor(D1).read()< 30) && (brick.sensor(D1).read()> 1))
{
    if (tNap == 0) {
        if (sector[tX*8+1+tY].left == 1) {

            sector[tX*8+1+tY].left = 0;
            ERR = 1;
            print("erF")
            brick.playSound("media/beep.wav");
            script.wait(300);

        }
    }
    if (tNap== 1) {
        if (sector[tX*8+1+tY].up == 1) {
            sector[tX*8+1+tY].up = 0;
            ERR = 1;
            print("erF")
            brick.playSound("media/beep.wav");
            script.wait(300);

        }
    }
    if (tNap== 2) {

```



```

var rotate = function(ar) {
    //run(20);
    var arc = ar;
    var k = 117;
    var e1 = brick.encoder(E3);
    var er = brick.encoder(E4);
    e1.reset();
    er.reset();// обнулили енкодеры
    mr=brick.motor(M4);
    ml=brick.motor(M3);
    var ob = 75 * Math.PI;
    if (arc < 0) {
        var leng = 2*k*Math.PI* -arc / 360;
        mr.setPower(60);
        ml.setPower(-60);
    }
    else {
        var leng = 2*k*Math.PI* arc / 360;
        mr.setPower(-60);
        ml.setPower(60);
    }
    var limit = (360/ob)*leng;
    var enc = 0;
    while (enc < limit) {
        enc=Math.abs(er.readRawData());
        script.wait(5);
    }
    mr.powerOff();
    ml.powerOff();
    script.wait(500);
}

var sector = new Array(64);
sector [1]= new Kletka(0,0 ,0,1,1,0);
sector [2]= new Kletka(1,0 ,1,0,1,0);
sector [3]= new Kletka(2,0 ,1,1,0,0);
sector [4]= new Kletka(3,0 ,0,0,1,0);
sector [5]= new Kletka(4,0 ,1,1,1,0);
sector [6]= new Kletka(5,0 ,1,0,0,0);
sector [7]= new Kletka(6,0 ,0,1,1,0);
sector [8]= new Kletka(7,0 ,1,0,0,0);
sector [9]= new Kletka(0,1 ,0,1,0,1);
sector [10]= new Kletka(1,1 ,0,0,0,0);
sector [11]= new Kletka(2,1 ,0,1,0,1);
sector [12]= new Kletka(3,1 ,0,0,0,0);
sector [13]= new Kletka(4,1 ,0,1,0,1);
sector [14]= new Kletka(5,1 ,0,0,0,0);
sector [15]= new Kletka(6,1 ,0,1,0,1);
sector [16]= new Kletka(7,1 ,0,0,0,0);
sector [17]= new Kletka(0,2 ,0,1,1,1);
sector [18]= new Kletka(1,2 ,1,0,1,0);
sector [19]= new Kletka(2,2 ,1,0,1,1);
sector [20]= new Kletka(3,2 ,1,0,1,0);
sector [21]= new Kletka(4,2 ,1,1,1,1);
sector [22]= new Kletka(5,2 ,1,0,1,0);
sector [23]= new Kletka(6,2 ,1,1,1,1);
sector [24]= new Kletka(7,2 ,1,1,0,0);
sector [25]= new Kletka(0,3 ,0,1,0,1);
sector [26]= new Kletka(1,3 ,0,0,1,0);
sector [27]= new Kletka(2,3 ,1,1,1,0);
sector [28]= new Kletka(3,3 ,1,0,0,0);

```

```
sector [29]= new Kletka(4,3 ,0,1,0,1);
sector [30]= new Kletka(5,3 ,0,0,0,0);
sector [31]= new Kletka(6,3 ,0,0,1,1);
sector [32]= new Kletka(7,3 ,1,1,0,1);
sector [33]= new Kletka(0,4 ,0,1,0,1);
sector [34]= new Kletka(1,4 ,0,0,0,0);
sector [35]= new Kletka(2,4 ,0,1,0,1);
sector [36]= new Kletka(3,4 ,0,0,0,0);
sector [37]= new Kletka(4,4 ,0,1,1,1);
sector [38]= new Kletka(5,4 ,1,1,0,0);
sector [39]= new Kletka(6,4 ,0,0,0,0);
sector [40]= new Kletka(7,4 ,0,1,0,1);
sector [41]= new Kletka(0,5 ,0,0,1,1);
sector [42]= new Kletka(1,5 ,1,1,1,0);
sector [43]= new Kletka(2,5 ,1,0,1,1);
sector [44]= new Kletka(3,5 ,1,1,1,0);
sector [45]= new Kletka(4,5 ,1,0,1,1);
sector [46]= new Kletka(5,5 ,1,1,1,1);
sector [47]= new Kletka(6,5 ,1,0,0,0);
sector [48]= new Kletka(7,5 ,0,1,0,1);
sector [49]= new Kletka(0,6 ,0,0,0,0);
sector [50]= new Kletka(1,6 ,0,1,0,1);
sector [51]= new Kletka(2,6 ,0,0,0,0);
sector [52]= new Kletka(3,6 ,0,1,0,1);
sector [53]= new Kletka(4,6 ,0,0,0,0);
sector [54]= new Kletka(5,6 ,0,1,0,1);
sector [55]= new Kletka(6,6 ,0,0,0,0);
sector [56]= new Kletka(7,6 ,0,1,0,1);
sector [57]= new Kletka(0,7 ,0,0,1,0);
sector [58]= new Kletka(1,7 ,1,0,0,1);
sector [59]= new Kletka(2,7 ,0,0,1,0);
sector [60]= new Kletka(3,7 ,1,0,1,1);
sector [61]= new Kletka(4,7 ,1,0,0,0);
sector [62]= new Kletka(5,7 ,0,0,1,1);
sector [63]= new Kletka(6,7 ,1,0,1,0);
sector [64]= new Kletka(7,7 ,1,0,0,1);
```

```
var klet = new Array();
var pos  = new Array(64);
var put  = new Array(64);
var napr = new Array(64);
var fir = 0;
var last = 0;
```

```
// var Text = script.readAll("input.txt");
// p1 = Text[0].indexOf(" ");
// p2 = Text[0].indexOf(" ",p1+1);
// p3 = Text[1].indexOf(" ");
// c1 = +Text[0].substring(0,p1);
// c2 = +Text[0].substring(p1+1,p2);
// c3 = +Text[0].substring(p2+1);
// c4 = +Text[1].substring(0,p3);
// c5 = +Text[1].substring(p3+1);
```

```
var tNap = 0;
var chik= 0;
var tX = 0;
var tY = 7;
var ENDX = 6;
var ENDY = 7;
```



```

var NEND = 3;
while ((tX !=ENDX) && (tY !=ENDY)) {
    deshifr (Shir(tX,tY,tNap,ENDX,ENDY));
}

var povorot = NEND - tNap;

if (Math.abs(povorot)> 2) {
    if (povorot > 0) {
        povorot -=4;
    } else {
        povorot+=4}
}

var pover = "";

if (povorot > 0) {
    for (i=0;i<povorot;i++) {
        pover = pover +"R"
    }
} else {
    for (i=0;i>povorot;i--) {
        pover = pover +"L"
    }
}

while (pover.length > 0) {
    vre = pover.substring(0,1);
    pover = pover.substring(1);
    if (vre=="L")
    {
        if ((brick.sensor(D1).read()< 25) &&
(brick.sensor(D1).read()> 1)) {
            brick.motor(M4).setPower(-60);
            brick.motor(M3).setPower(-60);

```

```

        script.wait(200);

        brick.motor(M4).powerOff();

        brick.motor(M3).powerOff();

    }

    rotate(-76)

    tNap=tNap-1;

}

if (vre=="R")

    {
        if ((brick.sensor(D1).read()< 25) &&
(brick.sensor(D1).read()> 1)) {

            brick.motor(M4).setPower(-60);

            brick.motor(M3).setPower(-60);

            script.wait(200);

            brick.motor(M4).powerOff();

            brick.motor(M3).powerOff();

        }

        tNap=tNap+1;

        rotate(76)

    }

}

Xk = 0;

Yk = 0;

while ((tX !=Xk) && (tY !=Yk)) {

    deshifr (Shir(tX,tY,tNap,Xk,Yk));

}

//XX - x координата финиша; YY - y координата финиша; tNap - направление
в данный момент;
// tX - текущая координата x; tY -текущая координата y;

//    proezd(XX,YY,tNap,7,7)

//        brick.playSound("media/beep.wav");
//        run(20);
//        script.wait(4999);
//        brick.playSound("media/beep.wav");

```

```

brick.motor(M3).setPower(-40);
brick.motor(M4).setPower(-40);
script.wait(300);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
if (brick.sensor(A2).read()<24){chik++}
brick.display().addLabel("finish",2,2);
brick.display().addLabel(chik,2,70);
brick.display().redraw();
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
script.wait(9999);

```

```

function Kletka(x,y,left,up,right,down,mark)
{
    this.x=x;
    this.y=y;
    this.left=left;
    this.up=up;
    this.right=right;
    this.down=down;
    this.mark=mark;
}

```

```

function fotog() {
    var photo = [];

    // перевести бинарное число в десятичное
    function count(num1, num2, num3, num4) { return (num1 *
Math.pow(2,3) + num2 * Math.pow(2,2) + num3 * Math.pow(2,1) + num4 *
Math.pow(2,0)); }

    // функция берёт сырое изображение и на выходе даёт в переменной
photo бинарное обрезанное изображение
    function reader_checker(raw_photo){
        n = 120;
        m = 159;
        var z = 0;
        var x_min, x_max, y_max, y_min;
        for (var i = 0;i < n; i++){
            photo[i] = [];
            z++;
            // бинаризация изображения
            for (var j = 0; j < m; j++ ){
                z++
                var temp = (((raw_photo[z] & 0xff0000) >> 18)
+ ((raw_photo[z] & 0xff00) >> 10) + ((raw_photo[z] & 0xff) >> 2));
                photo[i].push(temp < 40 ? 1 : 0);
            }
        }
    }
}

```

```

m = 155;
var i = 0;
var j = 0;
while((i < n)){
    if (photo[i][j] == 1){
        var xmin = i;
        break;
    }
    i += (++j/m == 1) ? 1: 0;
    j %= m;
}

i = n - 1;
j = 0;
while((i >= 0)){
    if (photo[i][j] == 1){
        var xmax = i;
        break;
    }
    i -= (++j/m == 1) ? 1: 0;
    j %= m;
}

i = 0;
j = 0;
while((i < m)){
    if (photo[j][i] == 1){
        ymin = i;
        break;
    }
    i += (++j/n == 1) ? 1: 0;
    j %= n;
}

i = m - 1;
j = 0;
while((i >= 0)){
    if (photo[j][i] == 1){
        ymax = i;
        break;
    }
    i -= (++j/n == 1) ? 1: 0;
    j %= n;
}

var x = 0;
var y = 0;
var temp_photo = [];
for (i = xmin; i <= xmax; i++){
    temp_photo[x] = [];
    for (var j = ymin; j <= ymax; j++){
        temp_photo[x][y] = photo[i][j];
        y++;
    }
    //print(temp_photo[x]);
    x++;
}

```

```

        y = 0;
    }
    photo = temp_photo;

    // новые размеры изображения
    n = xmax - xmin + 1;
    m = ymax - ymin + 1;
    // если не нашли маркер
    return ((n<2)||m<2) ? -1 : 1;
}

// находим среднюю точку между двумя точками
function find_middle_point(t_x1, t_y1, t_x2, t_y2){
    var t_output = [];
    t_output.push(Math.floor((t_x2 - t_x1) * 0.5 + t_x1));
    t_output.push(Math.floor((t_y2 - t_y1) * 0.5 + t_y1));
    return t_output;
}

function count_center(x1, x2, x3, x4, y1, y2, y3, y4) {
    var x, y;
    var a1 = y1 - y3;
    var b1 = x3 - x1;
    var c1 = -(a1 * x1 + b1 * y1);
    var a2 = y2 - y4;
    var b2 = x4 - x2;
    var c2 = -(a2 * x2 + b2 * y2);
    x = -(c1 * b2 - c2 * b1) / (a1 * b2 - a2 * b1);
    y = -(a1 * c2 - a2 * c1) / (a1 * b2 - a2 * b1);
    var t_output = [];
    t_output.push(Math.floor(x));
    t_output.push(Math.floor(y));
    return t_output;
}

function detect() {
    var x1, x2, x3, x4, y1, y2, y3, y4;

    var i = 0;
    var j = 0;
    var k = 0;
    while(i < n){
        if (photo[k][j] == 1){
            x1 = k;
            y1 = j;
            break;
        }
        i += (--k < 0) ? 1 : 0;
        k = (k < 0) ? i : k;
        j++;
        j %= m;
    }

    i = m - 1;
    j = 0;
    k = m - 1;
    while(i > 0){
        if (photo[j][k] == 1){
            x2 = j;
            y2 = k;

```

```

        break;
    }
    i -= (++k >= m) ? 1: 0;
    k = (k >= m) ? i : k;
    j++;
    j %= n;
}

i = n - 1;
k = n - 1;
j = m - 1;
while( i >= 0){
    if (photo[k][j] == 1){
        x3 = k;
        y3 = j;
        break;
    }

    i -= (++k >= n) ? 1: 0;
    k = (k >= n) ? i : k;
    j--;
    j = (j <0) ? m-1 : j;
}

i = 0;
k = 0;
j = n - 1;
while( i < m){
    if (photo[j][k] == 1){
        x4 = j;
        y4 = k;
        break;
    }

    i += (--k < 0) ? 1: 0;
    k = (k < 0) ? i : k;
    j--;
    j = (j <0) ? n-1 : j;
}

var center_x, center_y;
var res = [];
res = count_center(x1, x2, x3, x4, y1, y2, y3, y4);
center_x = res[0];
center_y = res[1];
var x_top = Math.floor((x1 + x2) / 2);
var y_top = Math.floor((y1 + y2) / 2);
var x_left = Math.floor((x1 + x4) / 2);
var y_left = Math.floor((y1 + y4) / 2);
var x_down = Math.floor((x3 + x4) / 2);
var y_down = Math.floor((y3 + y4) / 2);
var x_right = Math.floor((x2 + x3) / 2);
var y_right = Math.floor((y2 + y3) / 2);
var col = [];
col[0] = [];
col[0] = find_middle_point(center_x, center_y, x1, y1);
col[1] = [];

```

```

        col[1] = find_middle_point(center_x, center_y, x2, y2);
        col[2] = [];
        col[2] = find_middle_point(center_x, center_y, x3, y3);
        col[3] = [];
        col[3] = find_middle_point(center_x, center_y, x4, y4);

        var temp_number1 = find_middle_point(center_x, center_y,
x_top, y_top);
        var temp_number2 = find_middle_point(center_x, center_y,
x_left, y_left);
        var temp_number3 = find_middle_point(center_x, center_y,
x_right, y_right);
        var temp_number4 = find_middle_point(center_x, center_y,
x_down, y_down);

        if (photo[col[0][0]][col[0][1]] == 0)
            return
count(photo[temp_number4[0]][temp_number4[1]],
photo[temp_number3[0]][temp_number3[1]],
photo[temp_number2[0]][temp_number2[1]],
photo[temp_number1[0]][temp_number1[1]]);
        else if (photo[col[1][0]][col[1][1]] == 0)
            return
count(photo[temp_number2[0]][temp_number2[1]],
photo[temp_number4[0]][temp_number4[1]],
photo[temp_number1[0]][temp_number1[1]],
photo[temp_number3[0]][temp_number3[1]]);
        else if (photo[col[2][0]][col[2][1]] == 0)
            return
count(photo[temp_number1[0]][temp_number1[1]],
photo[temp_number2[0]][temp_number2[1]],
photo[temp_number3[0]][temp_number3[1]],
photo[temp_number4[0]][temp_number4[1]]);
        else
            return
count(photo[temp_number3[0]][temp_number3[1]],
photo[temp_number1[0]][temp_number1[1]],
photo[temp_number4[0]][temp_number4[1]],
photo[temp_number2[0]][temp_number2[1]]);
    }
    /*
function cameraDetect(_raw_photo){
    var error=reader_checker(_raw_photo);
    if (error>0)
    {counter = detect();
brick.display().addLabel(counter,10,10);
brick.display().redraw();
print(counter);
return counter
script.wait(5000);
}
}*/

var f_x_end = -1;
var f_y_end = -1

brick.configure("video1", "lineSensor");

// делаем три фото
var tml = reader_checker(getPhoto()) > 0 ? detect(): -1;
// print("tml "+tml);

```

```

brick.motor(M4).setPower(-75);
brick.motor(M3).setPower(-75);
script.wait(30);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
var tm2 = reader_checker(getPhoto()) > 0 ? detect(): -1;
// print("tm2 "+tm2);
brick.motor(M4).setPower(75);
brick.motor(M3).setPower(75);
script.wait(60);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
var tm3 = reader_checker(getPhoto()) > 0 ? detect(): -1;
// print("tm3 "+tm3);
brick.motor(M4).setPower(-75);
brick.motor(M3).setPower(-75);
script.wait(30);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
// делаем три фото

var temp_coord =0;

if ((tm1==tm2) || (tm1==tm3))
{
    temp_coord = tm1;
}
if ((tm2==tm3) && (tm2!=tm1))
{
    temp_coord = tm2;
}
if ((tm2!=tm3) && (tm2!=tm1)&& (tm1!=tm3))
{
    temp_coord = tm3;
}

brick.motor(M4).setPower(-75);
brick.motor(M3).setPower(-70);
script.wait(530);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();

script.wait(1000);

if (temp_coord > 7)
{
    var tm1 = reader_checker(getPhoto()) > 0 ? detect():
-1;
    // print("tm1 "+tm1);
    brick.motor(M4).setPower(-75);
    brick.motor(M3).setPower(-75);
    script.wait(20);
    brick.motor(M4).powerOff();
    brick.motor(M3).powerOff();
    var tm2 = reader_checker(getPhoto()) > 0 ? detect():
-1;
    // print("tm2 "+tm2);
    brick.motor(M4).setPower(75);

```



```

brick.motor(M3).setPower(75);
script.wait(40);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
var tm3 = reader_checker(getPhoto()) > 0 ? detect():
-1;
// print("tm3 "+tm3);
brick.motor(M4).setPower(-75);
brick.motor(M3).setPower(-75);
script.wait(20);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
if ((tm1==tm2) || (tm1==tm3))
{
    f_x_end = tm1;
}
if ((tm2==tm3) && (tm2!=tm1))
{
    f_x_end = tm2;
}
if ((tm2!=tm3) && (tm2!=tm1)&& (tm1!=tm3))
{
    f_x_end = tm1;
}
f_y_end = temp_coord - 8;
}else{
f_x_end = temp_coord;
var tm1 = reader_checker(getPhoto()) > 0 ? detect():
-1;
// print("tm1 "+tm1);
brick.motor(M4).setPower(-75);
brick.motor(M3).setPower(-75);
script.wait(20);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
var tm2 = reader_checker(getPhoto()) > 0 ? detect():
-1;
// print("tm2 "+tm2);
brick.motor(M4).setPower(75);
brick.motor(M3).setPower(75);
script.wait(40);
brick.motor(M4).powerOff();
brick.motor(M3).powerOff();
var tm3 = reader_checker(getPhoto()) > 0 ? detect():
-1;
// print("tm3 "+tm3);
brick.motor(M4).setPower(-75);
brick.motor(M3).setPower(-75);
script.wait(20);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
if ((tm1==tm2) || (tm1==tm3))
{
    f_y_end = tm1-8;
}
if ((tm2==tm3) && (tm2!=tm1))
{
    f_y_end = tm2-8;
}
if ((tm2!=tm3) && (tm2!=tm1)&& (tm1!=tm3))
{

```

```

        f_y_end = tm3-8;
    }
}
Xk = f_x_end;
Yk = f_y_end;
//print(f_x_end + " " + f_y_end);
brick.display().clear();
brick.display().addLabel(f_x_end + " " + f_y_end,100,10);
brick.display().redraw();
run(200);
script.wait(150);
}

// выдает кратчайший путь от точки А до точки В в формате "FLR"
function Shir(y1,x1,V,y2,x2)
{
    fir =1;
    last =1;
    for(var i = 1;i<65;i++)
    {
        pos[i]=0;
    };
    for(var i = 1;i<65;i++)
    {
        put[i]="";
    };
    for(var i = 1;i<65;i++)
    {
        napr[i]=0;
    }
    for(var i = 1;i<300;i++)
    {
        klet[i]= 0;
    }

    var S= x1+1 + y1*8;
    var F= x2+1 + y2*8;
    napr[S]=V;
    add(S);
    while (true) {
        ost = take();
        if (ost == F) {
            break;
        };
        if (sector[ost].left == 1) {
            if (napr[ost] == 0) {
                K = put[ost]+"F";
            };
            if (napr[ost] == 1) {
                K = put[ost]+"L"+"F";
            };
            if (napr[ost] == 2) {
                K = put[ost]+"LL"+"F";
            };
            if (napr[ost] == 3) {
                K = put[ost]+"R"+"F";
            };
            if (pos[ost-1] == 0) {
                put[ost-1] = K;
                add (ost-1);
            }
        }
    }
}

```

```

//          napr[ost-1] = 0;
//      } else {
//          if (K < put[ost-1]) {
//              put[ost-1] = K;
//              add (ost-1);
//              napr[ost-1] = 0;
//          }
//      }
};
if (sector[ost].up == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"R"+"F";
    };
    if (napr[ost] == 1) {
        K = put[ost]+"F";
    };
    if (napr[ost] == 2) {
        K = put[ost]+"L"+"F";
    };
    if (napr[ost] == 3) {
        K = put[ost]+"LL"+"F";
    };
    if (pos[ost+8] == 0) {
        put[ost+8] = K;
        add (ost+8);
        napr[ost+8] = 1;
//      } else {
//          if (K < put[ost+8]) {
//              put[ost+8] = K;
//              add (ost+8);
//              napr[ost+8] = 1;
//          }
//      }
    };
if (sector[ost].right == 1) {
    if (napr[ost] == 0) {
        K = put[ost]+"LL"+"F";
    };
    if (napr[ost] == 1) {
        K = put[ost]+"R"+"F";
    };
    if (napr[ost] == 2) {
        K = put[ost]+"F";
    };
    if (napr[ost] == 3) {
        K = put[ost]+"L"+"F";
    };
    if (pos[ost+1] == 0) {
        put[ost+1] = K;
        add (ost+1);
        napr[ost+1] = 2;
//      } else {
//          if (K < put[ost+1]) {
//              put[ost+1] = K;
//              add (ost+1);
//              napr[ost+1] = 2;
//          }
//      }
    };
if (sector[ost].down == 1) {
    if (napr[ost] == 0) {

```

```

        K = put[ost]+"L"+"F";
    }
    if (napr[ost] == 1) {
        K = put[ost]+"LL"+"F";
    }
    if (napr[ost] == 2) {
        K = put[ost]+"R"+"F";
    }
    if (napr[ost] == 3) {
        K = put[ost]+"F";
    }
    if (pos[ost-8] == 0) {
        put[ost-8] = K;
        add (ost-8);
        napr[ost-8] = 3;
    } else {
        if (K < put[ost-8]) {
            put[ost-8] = K;
            add (ost-8);
            napr[ost-8] =3;
        }
    }
}
if (length() == 0) {
    return 0;
}
}
return put[F];
}

function add(q) {
    klet[last]=q;
    last=last+1;
    pos[q]=1;
};

function length() {
    return last-fir;
};

function take() {
    fir=fir+1;
    return klet[fir-1];
}
var run = function(length) {
    var leng = length;
    var p =60;
    var ob = 75 * Math.PI;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили енкодеры

    mr=brick.motor(M4);
    ml=brick.motor(M3);
    mr.setPower(p);
    ml.setPower(p);//запустили моторы

    var limit = (374/ob)*leng;

```

```

var enc = 0;
var k = 1;
var D = 1;
var df = 2;
var l = 4;
var razn = 0;
var razn_old = 0;
while (enc < limit) {
    if (brick.sensor(A2).read() < 18) { //езда по правой руке
        razn = (brick.sensor(A2).read()-14);
        g = razn*l + (razn-razn_old)*df;
        mr.setPower(p-g);
        ml.setPower(p+g);
        razn_old=razn;
    } else {
        if (brick.sensor(A3).read() < 18) { //езда по левой
руке
            razn = (brick.sensor(A3).read()-14);
            g = razn*l + (razn-razn_old)*df;
            mr.setPower(p+g);
            ml.setPower(p-g);
            razn_old=razn;
        }
    }
    if ((brick.sensor(A2).read() >
18)&&(brick.sensor(A3).read() > 18)) { //езда вперед
        razn = (Math.abs(el.readRawData()) -
Math.abs(er.readRawData()))
        g = razn*k + (razn-razn_old)*D;
        mr.setPower(p+g);
        ml.setPower(p-g);
        razn_old=razn;
    }
    enc=(Math.abs(el.readRawData()) +
Math.abs(er.readRawData())) / 2;
    script.wait(10);
}
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
script.wait(500);
}

// доезжает от точки А до точки В (если завал, то останавливается)
function deshifr(str) {
    brick.motor(M3).powerOff();
    brick.motor(M4).powerOff();
    while (str.length > 0)
    {
        if (tNap > 0) {
            tNap= tNap % 4
        }
        if (tNap < 0) {
            tNap = 4 - Math.abs(tNap % 4)
        }
    }
    sim = str.substring(0,1);
    str = str.substring(1);
    ERR=0;
    // поиск завала передним датчиком

```

```

if ((brick.sensor(D1).read() < 38) && (brick.sensor(D1).read() > 1))
{
    if (tNap == 0) {
        if (sector[tX*8+1+tY].left == 1) {
            if (tY < 8) {sector[tX*8+1+tY].left = 0;}
            if (tY > 1) { sector[tX*8+1+tY - 2].right =
0}

            if (tX > 0) {sector[tX*8+1+tY - 9].up = 0}
            if (tX < 7) {sector[tX*8+1+tY + 7].down = 0}

            ERR = 1;
            print("erF")
            brick.playSound("media/beep.wav");
            script.wait(500);
        }
    }
    if (tNap == 1) {
        if (sector[tX*8+1+tY].up == 1) {
            if (tY < 7) {sector[tX*8+1+tY + 9].left = 0;}
            if (tY > 0) { sector[tX*8+1+tY + 7].right =
0}

            if (tX > -1) {sector[tX*8+1+tY].up = 0}
            if (tX < 6) {sector[tX*8+1+tY + 16].down = 0}
            ERR = 1;
            print("erF")
            brick.playSound("media/beep.wav");
            script.wait(500);
        }
    }
    if (tNap == 2) {
        if (sector[tX*8+1+tY].right == 1) {
            if (tY < 6) {sector[tX*8+1+tY + 2].left = 0;}
            if (tY > -1) { sector[tX*8+1+tY].right = 0}
            if (tX > 0) {sector[tX*8+1+tY-7].up = 0}
            if (tX < 7) {sector[tX*8+1+tY + 9].down = 0}
            ERR = 1;
            print("erF")
            brick.playSound("media/beep.wav");
            script.wait(500);
        }
    }
    if (tNap == 3) {
        if (sector[tX*8+1+tY].down == 1) {
            if (tY < 7) {sector[tX*8+1+tY - 7].left = 0;}
            if (tY > 0) { sector[tX*8+1+tY - 9].right =
0}

            if (tX > 1) {sector[tX*8+1+tY - 16].up = 0}
            if (tX < 8) {sector[tX*8+1+tY].down = 0}
            ERR = 1;
            print("erF")
            brick.playSound("media/beep.wav");
            script.wait(500);
        }
    }
};

```

```

//если завал то остановись
if (ERR == 1) {
    ERROR +=1;
}

```

```

        //      for(i=1;i < 65;i++) {
        //          print(i+"      "+sector[i].left + " "+sector[i].up +
" "+sector[i].right + " "+sector[i].down + " ")
        //      }
        brick.display().clear();
        Z1 =zabl()
        print(Z1)
        brick.display().addLabel(zabl(),2,2);
        brick.display().redraw();
        script.wait(1000);
        break;
    }

    // расшифровка кода и проезд

    if (sim=="F")
    {print("f")
        run(530);
        if (tNap == 0) {tY=tY-1};
        if (tNap == 1) {tX=tX+1};
        if (tNap == 2) {tY=tY+1};
        if (tNap == 3) {tX=tX-1};
    }
    if (sim=="L")
    {print("l")
        if (brick.sensor(D1).read()<5) {
            brick.motor(M4).setPower(-60);
            brick.motor(M3).setPower(-60);
            script.wait(100);
            brick.motor(M4).powerOff();
            brick.motor(M3).powerOff();
        }
        rotate(-80)
        tNap=tNap-1;
    }
    if (sim=="R")
    {
        print("r")
        if (brick.sensor(D1).read()<5) {
            brick.motor(M4).setPower(-60);
            brick.motor(M3).setPower(-60);
            script.wait(100);
            brick.motor(M4).powerOff();
            brick.motor(M3).powerOff();
        }
        tNap=tNap+1;
        rotate(80)
    }
    sector[tX*8+1+tY].mark=1;
}

var rotate = function(ar) {
    //run(20);
    var arc = ar;
    var k = 117;
    var el = brick.encoder(E3);
    var er = brick.encoder(E4);
    el.reset();
    er.reset();// обнулили энкодеры
}

```

```

mr=brick.motor(M4);
ml=brick.motor(M3);
var ob = 75 * Math.PI;
if (arc < 0) {
    var leng = 2*k*Math.PI* -arc / 360;
    mr.setPower(60);
    ml.setPower(-60);
}
else {
    var leng = 2*k*Math.PI* arc / 360;
    mr.setPower(-60);
    ml.setPower(60);
}
var limit = (360/ob)*leng;
var enc = 0;
while (enc < limit) {
    enc=Math.abs(er.readRawData());
    script.wait(5);
}
mr.powerOff();
ml.powerOff();
script.wait(500);
}

function zabl() {
    var zabl = 0;
    for (var i=0;i<8;i++) {
        for (var j=0;j<8;j++) {
            if ( Shir(tX,tY,tNap,i,j) == 0) {
                zabl +=1;
            }
        }
    }
}
return zabl-13;
}

// объявление лабиринта (x,у,лево,верх,право,низ)
var sector = new Array(64);
sector [1]= new Kletka(0,0 ,0,1,1,0, 0);
sector [2]= new Kletka(1,0 ,1,0,1,0, 0);
sector [3]= new Kletka(2,0 ,1,1,0,0, 0);
sector [4]= new Kletka(3,0 ,0,0,1,0, 0);
sector [5]= new Kletka(4,0 ,1,1,1,0, 0);
sector [6]= new Kletka(5,0 ,1,0,0,0, 0);
sector [7]= new Kletka(6,0 ,0,1,1,0, 0);
sector [8]= new Kletka(7,0 ,1,0,0,0, 0);
sector [9]= new Kletka(0,1 ,0,1,0,1, 0);
sector [10]= new Kletka(1,1 ,0,0,0,0, 0);
sector [11]= new Kletka(2,1 ,0,1,0,1, 0);
sector [12]= new Kletka(3,1 ,0,0,0,0, 0);
sector [13]= new Kletka(4,1 ,0,1,0,1, 0);
sector [14]= new Kletka(5,1 ,0,0,0,0, 0);
sector [15]= new Kletka(6,1 ,0,1,0,1, 0);
sector [16]= new Kletka(7,1 ,0,0,0,0, 0);
sector [17]= new Kletka(0,2 ,0,1,1,1, 0);
sector [18]= new Kletka(1,2 ,1,0,1,0, 0);
sector [19]= new Kletka(2,2 ,1,0,1,1, 0);
sector [20]= new Kletka(3,2 ,1,0,1,0, 0);
sector [21]= new Kletka(4,2 ,1,1,1,1, 0);
sector [22]= new Kletka(5,2 ,1,0,1,0, 0);
sector [23]= new Kletka(6,2 ,1,1,1,1, 0);

```



```
sector [24]= new Kletka(7,2 ,1,1,0,0, 0);
sector [25]= new Kletka(0,3 ,0,1,0,1, 0);
sector [26]= new Kletka(1,3 ,0,0,1,0, 0);
sector [27]= new Kletka(2,3 ,1,1,1,0, 0);
sector [28]= new Kletka(3,3 ,1,0,0,0, 0);
sector [29]= new Kletka(4,3 ,0,1,0,1, 0);
sector [30]= new Kletka(5,3 ,0,0,0,0, 0);
sector [31]= new Kletka(6,3 ,0,0,1,1, 0);
sector [32]= new Kletka(7,3 ,1,1,0,1, 0);
sector [33]= new Kletka(0,4 ,0,1,0,1, 0);
sector [34]= new Kletka(1,4 ,0,0,0,0, 0);
sector [35]= new Kletka(2,4 ,0,1,0,1, 0);
sector [36]= new Kletka(3,4 ,0,0,0,0, 0);
sector [37]= new Kletka(4,4 ,0,1,1,1, 0);
sector [38]= new Kletka(5,4 ,1,1,0,0, 0);
sector [39]= new Kletka(6,4 ,0,0,0,0, 0);
sector [40]= new Kletka(7,4 ,0,1,0,1, 0);
sector [41]= new Kletka(0,5 ,0,0,1,1, 0);
sector [42]= new Kletka(1,5 ,1,1,1,0, 0);
sector [43]= new Kletka(2,5 ,1,0,1,1, 0);
sector [44]= new Kletka(3,5 ,1,1,1,0, 0);
sector [45]= new Kletka(4,5 ,1,0,1,1, 0);
sector [46]= new Kletka(5,5 ,1,1,1,1, 0);
sector [47]= new Kletka(6,5 ,1,0,0,0, 0);
sector [48]= new Kletka(7,5 ,0,1,0,1, 0);
sector [49]= new Kletka(0,6 ,0,0,0,0, 0);
sector [50]= new Kletka(1,6 ,0,1,0,1, 0);
sector [51]= new Kletka(2,6 ,0,0,0,0, 0);
sector [52]= new Kletka(3,6 ,0,1,0,1, 0);
sector [53]= new Kletka(4,6 ,0,0,0,0, 0);
sector [54]= new Kletka(5,6 ,0,1,0,1, 0);
sector [55]= new Kletka(6,6 ,0,0,0,0, 0);
sector [56]= new Kletka(7,6 ,0,1,0,1, 0);
sector [57]= new Kletka(0,7 ,0,0,1,0, 0);
sector [58]= new Kletka(1,7 ,1,0,0,1, 0);
sector [59]= new Kletka(2,7 ,0,0,1,0, 0);
sector [60]= new Kletka(3,7 ,1,0,1,1, 0);
sector [61]= new Kletka(4,7 ,1,0,0,0, 0);
sector [62]= new Kletka(5,7 ,0,0,1,1, 0);
sector [63]= new Kletka(6,7 ,1,0,1,0, 0);
sector [64]= new Kletka(7,7 ,1,0,0,1, 0);
```

```
var klet = new Array();
var pos  = new Array(64);
var put  = new Array(64);
var napr = new Array(64);
var fir = 0;
var last = 0; // объявление служебных переменных;
```

```
// var Text = script.readAll("input.txt");           <--- мусор
// p1 = Text[0].indexOf(" ");
// p2 = Text[0].indexOf(" ",p1+1);
// p3 = Text[1].indexOf(" ");                       <--- мусор
// c1 = +Text[0].substring(0,p1);
// c2 = +Text[0].substring(p1+1,p2);               <--- мусор
// c3 = +Text[0].substring(p2+1);
// c4 = +Text[1].substring(0,p3);
// c5 = +Text[1].substring(p3+1);                   <--- мусор
```

```

// объявление начальных переменных
var tNap = 3; // направление старта
var tX = 7; // x старта
var tY = 1; // y старта
var ENX = 5 // x artag
var ENY = 6; // y artag
var NEND = 3; // направление в которое мы должны стать, чтобы увидеть
artag
var Xk; // x финиша
var Yk; // y Финиша
var ERROR = 0; // количество завалов

// путь до artag, с поиском завалов
while (true) {
    if ((tX == ENX) && (tY == ENY)) { break}
    var tem = Shir(tX,tY,tNap,ENX,ENY)
    deshifr(tem)
}

// поворот к artag
var povorot = NEND - tNap;
if (Math.abs(povorot) > 2) {
    if (povorot > 0) {
        povorot -=4;
    } else {
        povorot +=4}
}
//=====
// поворот к artag // если не работает переписать !!!!!
var pover = "";
if (povorot > 0) {
    for (i=0;i<povorot;i++) {
        pover = pover + "R"
    }
} else {
    for (i=0;i>povorot;i--) {
        pover = pover + "L"
    }
}

// поворот к artag , можно было не кодировать а сразу
поворачиваться
while (pover.length > 0) {
    vre = pover.substring(0,1);
    pover = pover.substring(1);
    if (vre=="L")
    {
        if ((brick.sensor(D1).read() < 25) &&
(brick.sensor(D1).read() > 1)) {
            brick.motor(M4).setPower(-60);
            brick.motor(M3).setPower(-60);
            script.wait(200);
            brick.motor(M4).powerOff();
            brick.motor(M3).powerOff();
            script.wait(200);
        }
        rotate(-83)
        tNap=tNap-1;
    }
    if (vre=="R")

```

```

        {          if ((brick.sensor(D1).read()< 25) &&
(brick.sensor(D1).read()> 1)) {
                                brick.motor(M4).setPower(-60);
                                brick.motor(M3).setPower(-60);
                                script.wait(200);
                                brick.motor(M4).powerOff();
                                brick.motor(M3).powerOff();
                                script.wait(200);
                                }
                                tNap=tNap+1;
                                rotate(83)
        }
    }
}

//=====
//считывание artag
brick.playSound("media/beep.wav");
fotog();

//тут будет поиск оставшихся завалов
while (ERROR < 2) {
    print("----"+ERROR)
    mindlin="a";
    var tm;
    for (var i=0;i<8;i++) {
        for (var j=0;j<8;j++) {
            if ((sector[i*8+1+j].mark!=1) &&
(sector[i*8+1+j].mark!=2)) {
                tm = Shir(tX,tY,tNap,i,j);
//                print(tm)
                if (tm == 0) {
                    sector[i*8+1+j].mark=2;
                } else {
                    if (tm < mindlin) {
                        mindlin=tm;
                        script.wait(10);
                    }
                }
            }
        }
    }
    //    print(mindlin);
    sector[tX*8+1+tY].mark=1;
    deshifr(mindlin);
}

//остался поиск завалов;

//доезд до финиша
while (true) {
    if ((tX ==Xk) && (tY ==Yk)) {
        break;
    }
    deshifr (Shir(tX,tY,tNap,Xk,Yk));
}

//XX - x координата финиша; YY - y координата финиша; tNap - направление
в данный момент;
// tX - текущая координата x; tY -текущая координата y;

//    proezd(XX,YY,tNap,7,7)

```

```
//          brick.playSound("media/beep.wav");
//          run(20);
//          script.wait(4999);
//          brick.playSound("media/beep.wav");

brick.motor(M3).setPower(-40);
brick.motor(M4).setPower(-40);
script.wait(300);
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
brick.display().clear();
brick.display().addLabel("finish",2,2);
brick.display().addLabel(zabl(),20,20);
brick.display().redraw();
brick.motor(M3).powerOff();
brick.motor(M4).powerOff();
script.wait(9999);
```

Программы для решения полной финальной задачи нет.