function TForm1.Dekstra():string;

var

a:array[1..n,1..n] of longint;

b:array[1..n]of boolean;

d:array[1..n] of longint;

p:array[1..n] of string;

q, e, v, i, j, m: integer;

begin

q := StrToInt(Edit1.Text);

e := StrToInt(Edit2.Text);

if (q < 1) or (q > n) or (e < 1) or (e > n) then

raise ERangeError.Create('Значениедолжнобытьвдиапазоне 1..'+IntTostr(n));

for i := 1 to n do

begin

d[i]:=MaxInt;

for j := 1 to n do

a[j,i] := StrToIntDef(StringGrid1.Cells[i, j], -1);

end;

fillchar(b,sizeof(b),0);

d[q] := 0

p[q] := IntToStr(q);

v:=q;

repeat

b[v]:=true;

m:=MaxInt;

j:=-1;

for i:=1 to n do

if not b[i] then

begin

if (a[v,i] >= 0) and (d[v]+a[v,i] < d[i]) then

begin

d[i]:=d[v]+a[v,i];

p[i]:=Format('%s-%d',[p[v],i]);

end;

if d[i] < m then

begin

m:=d[i];

j:=i;

end;

end;

v:=j;

until (v < 0) or (v = e); //все пути пройдены или нашли минимальное расстояние для e

//выводрез-та

if d[e] = MaxInt then

ListBox1.Items.Add(Format('Путь из %d в %d не существует',[q,e]))

else

ListBox1.Items.Add(Format('Кратчайший путь из %d в %d: %s'#13#10'Длина пути: %d',[q,e,p[e],d[e]]));

result := p[e];

end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

nodeCount := nodeCount + 1;

StringGrid1.Cells[0, nodeCount] := IntToStr(nodeCount);

StringGrid1.Cells[nodeCount, 0] := IntToStr(nodeCount);

pointCoordinates[nodeCount - 1] := Point(X + 10,Y + 10);

image1.Canvas.Ellipse(X, Y, X + r, Y + r);

image1.Canvas.TextOut(X + 7,Y - 15,IntToStr(nodeCount))

end;

procedure TForm1.DrawPath;

var i,j:integer;

begin

Image1.Canvas.Pen.Color:=clBlack;

Image1.Canvas.Pen.Width:=2;

for i := 0 to NodeCount - 1 do

for j := 0 to NodeCount - 1 do

begin

if (i <> j) and (StringGrid1.Cells[i + 1, j + 1] <> '') then

begin

Image1.Canvas.MoveTo(pointCoordinates[i].X,pointCoordinates[i].Y);

Image1.Canvas.LineTo(pointCoordinates[j].X,pointCoordinates[j].Y);

DrawArrow(Image1.Canvas,pointCoordinates[i].X,pointCoordinates[i].Y,pointCoordinates[j].X,pointCoordinates[j].Y,3);

end;

end;

end;

procedure TForm1.DrawArrowHead(Canvas: TCanvas; X,Y: Integer; Angle,LW: Extended);

var

A1,A2: Extended;

Arrow: array[0..3] of TPoint;

OldWidth: Integer;

const

Beta=0.322;

LineLen=4.74;

CentLen=3;

begin

Angle:=Pi+Angle;

Arrow[0]:=Point(X,Y);

A1:=Angle-Beta;

A2:=Angle+Beta;

Arrow[1]:=Point(X+Round(LineLen\*LW\*Cos(A1)),Y-Round(LineLen\*LW\*Sin(A1)));

Arrow[2]:=Point(X+Round(CentLen\*LW\*Cos(Angle)),Y-Round(CentLen\*LW\*Sin(Angle)));

Arrow[3]:=Point(X+Round(LineLen\*LW\*Cos(A2)),Y-Round(LineLen\*LW\*Sin(A2)));

OldWidth:=Canvas.Pen.Width;

Canvas.Pen.Width:=1;

Canvas.Polygon(Arrow);

Canvas.Pen.Width:=OldWidth

end;

procedure TForm1.DrawArrow(Canvas: TCanvas; X1,Y1,X2,Y2: Integer; LW: Extended);

var

Angle: Extended;

begin

Angle:=ArcTan2(Y1-Y2,X2-X1);

Canvas.MoveTo(X1,Y1);

Canvas.LineTo(X2-Round(2\*LW\*Cos(Angle)),Y2+Round(2\*LW\*Sin(Angle)));

DrawArrowHead(Canvas,X2,Y2,Angle,LW);

end;

end.