Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**О Т Ч Е Т**

**по творческой работе**

**по дисциплине**

«Основы алгоритмизации и программирования» семестр 2

Выполнил студент гр. ИВТ-21-2б

Лебедев Тимофей Антонович

Проверил: Яруллин Денис Владимирович

(оценка) (подпись)

(дата)

г. Пермь-2022

**Постановка задачи:**

⦁ Реализовать вывод матрицы смежности и всех решений;

⦁ Реализовать функции для решения задачи методом ветвей и границ;

⦁ Реализовать нахождение кратчайшего пути;

⦁ Вывести путь и расстояние, которое по нему было пройдено;

⦁ Реализовать отрисовку графа с помощью SFML.

**Описание решения:**

Создание программы осуществлялось на основе метода ветвей и границ, для отрисовки графа был использован SFML, в ходе работы пришлось изучить инструмент SFML для отрисовки графа. Трудности возникли со скачиванием и установкой библиотеки SFML, но были преодолены.

В задаче Коммивояжера я горжусь понятной визуализацией графа, отображения пути и результата, которые были посчитаны и выведены верно, а также возможностью закрытия окна с помощью кнопки esc.

**Скриншоты:**

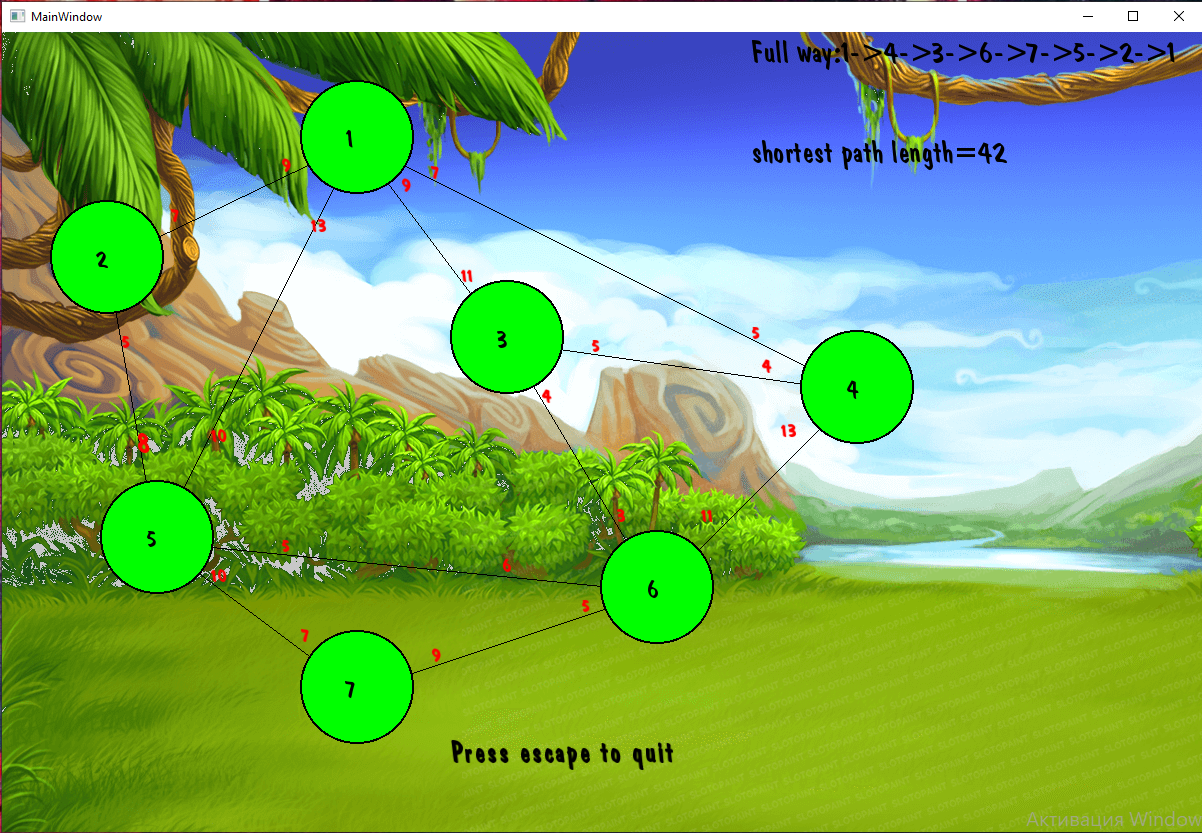
****

Рис 4. Визуализация графа через SFML

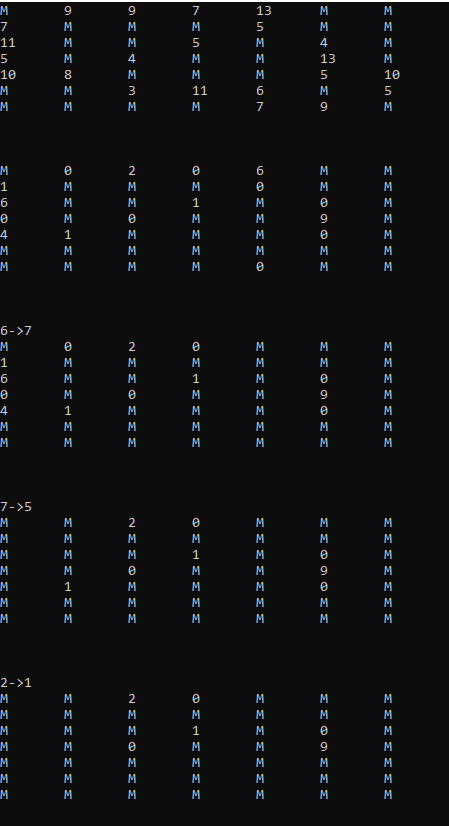


Рис 7. Вывод решения в консоль

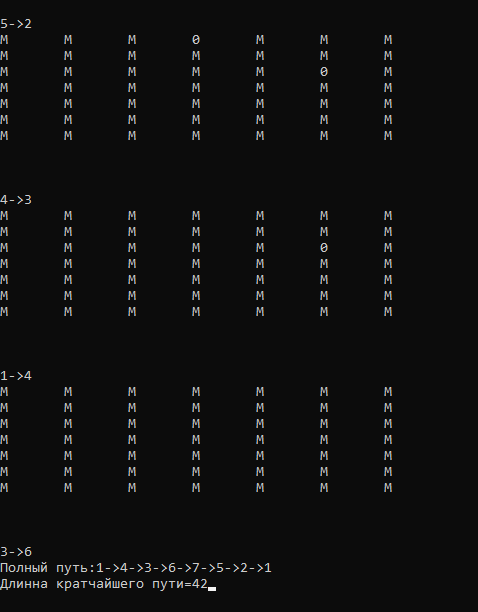
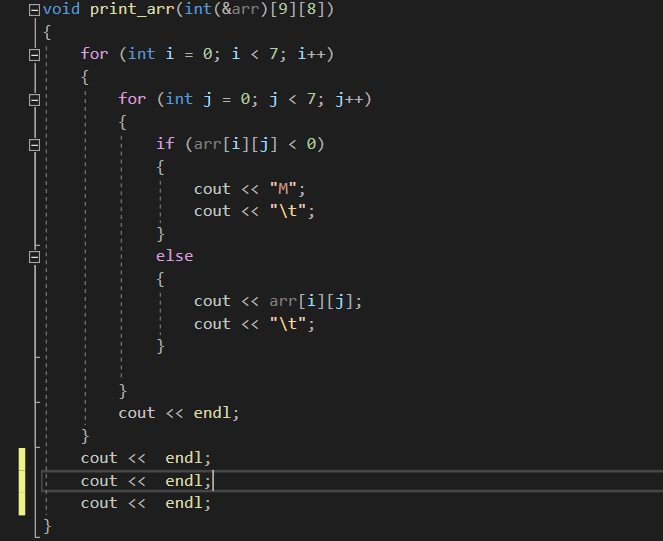


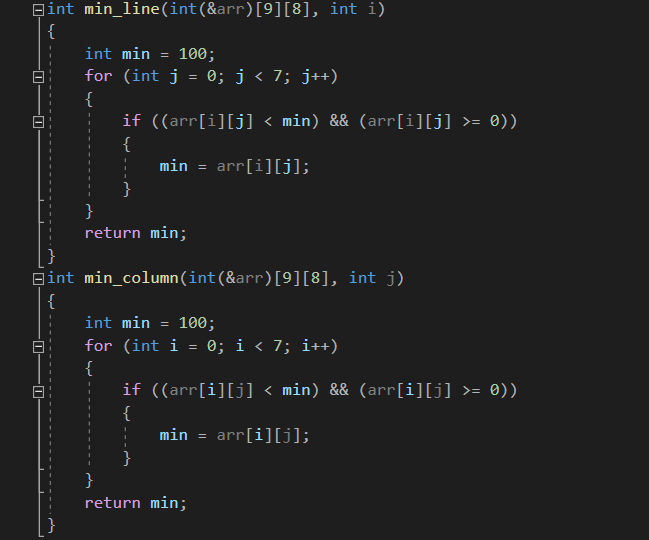
Рис 8. Вывод решения в консоль

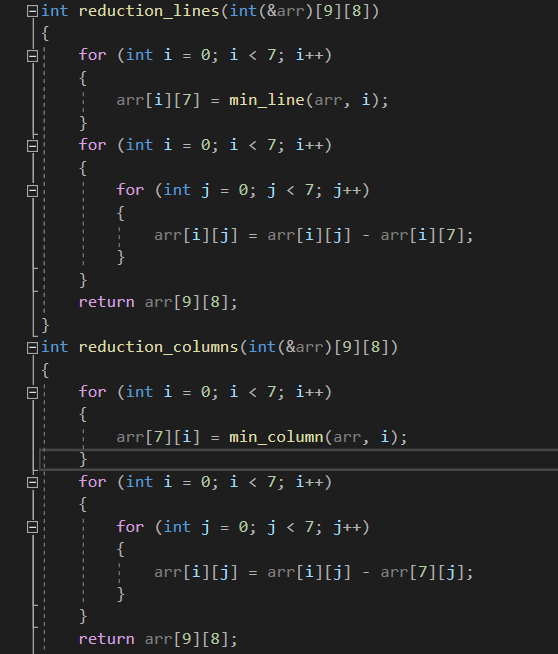
**Описание кода:**

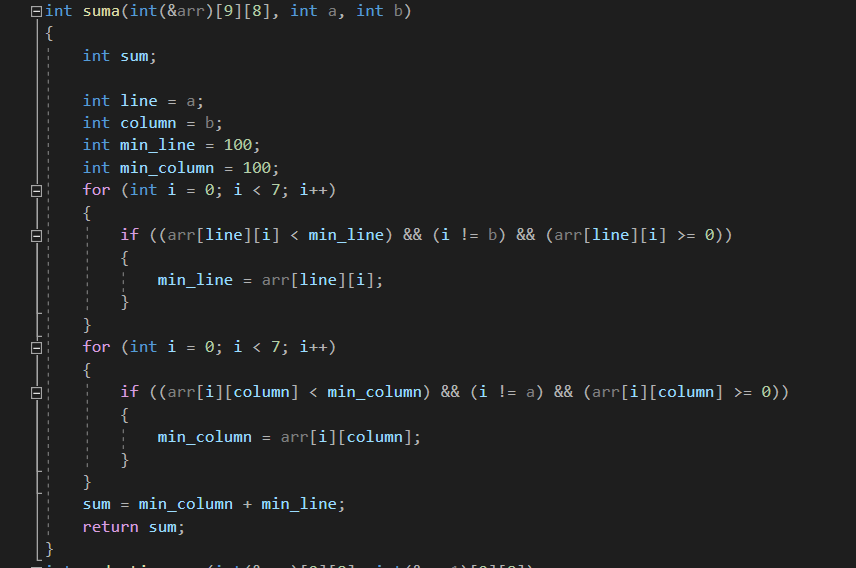
⦁ Вывод графа

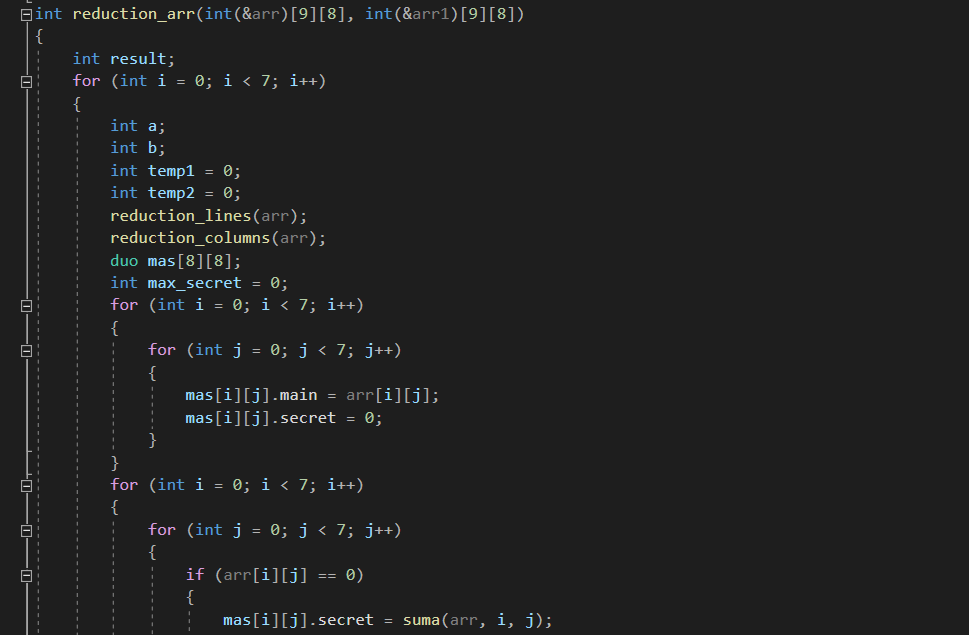


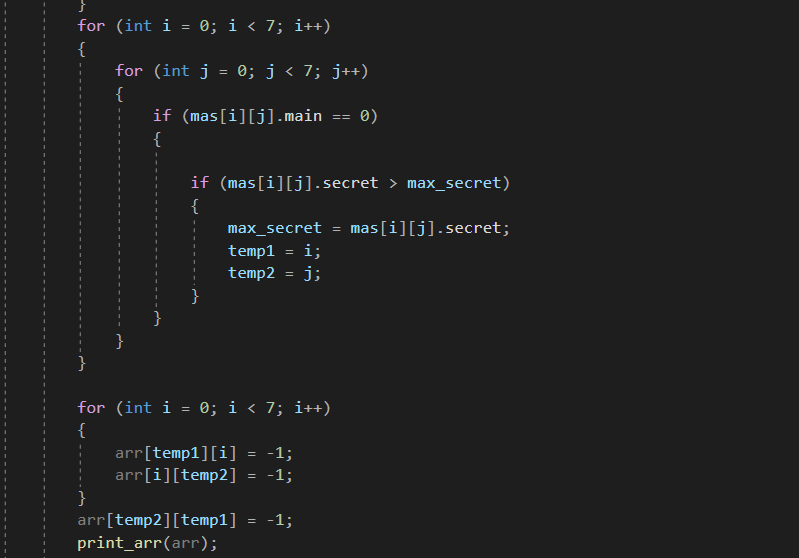
⦁ Выполнение метода ветвей и границ для поиска пути и расстояния

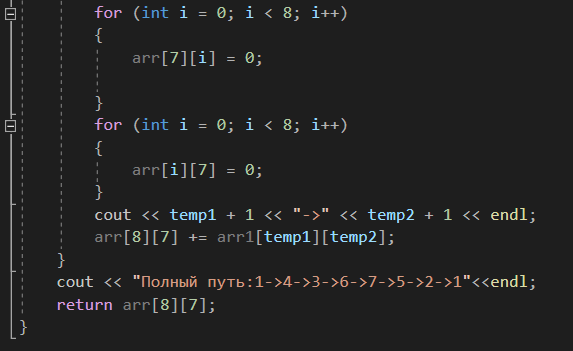




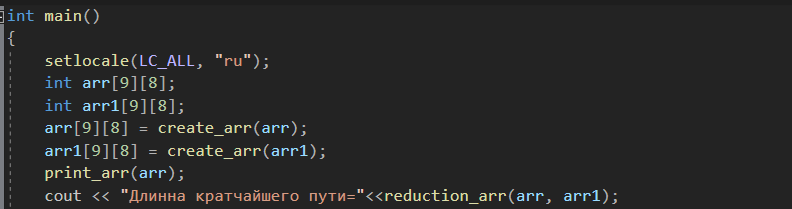








⦁ Функция main



⦁ Отрисовка графа

sf::RenderWindow window(sf::VideoMode(1200, 800), "MainWindow");

sf::Image image;

image.loadFromFile("ja.png");

image.createMaskFromColor(image.getPixel(0, 0));

sf::Texture texture;

texture.loadFromImage(image);

sf::Sprite sprite;

sprite.setTexture(texture);

int a2, b2;

a2 = 50;

b2 = 170;

sf::Font font;

font.loadFromFile("ofont.ru\_a\_DomInoRevObl.ttf");

sf::Text text2("2", font);

text2.setFillColor(sf::Color::Black);

text2.setPosition(a2+45, b2+40);

text2.setCharacterSize(25);

CircleShape circle2;

CircleShape circle1;

circle2.setRadius(55);

circle2.setFillColor(Color::Green);

circle2.setPosition(a2, b2);

circle2.setOutlineColor(Color::Black);

circle2.setOutlineThickness(2);

circle2.setPointCount(150);

int a1, b1;

a1 = 300;

b1 = 50;

sf::Text text1("1", font);

text1.setFillColor(sf::Color::Black);

text1.setPosition(a1 + 45, b1 + 40);

text1.setCharacterSize(25);

circle1.setRadius(55);

circle1.setFillColor(Color::Green);

circle1.setPosition(a1, b1);

circle1.setOutlineColor(Color::Black);

circle1.setOutlineThickness(2);

circle1.setPointCount(150);

int a5, b5;

a5 = 100;

b5 = 450;

sf::Text text5("5", font);

text5.setFillColor(sf::Color::Black);

text5.setPosition(a5 + 45, b5 + 40);

text5.setCharacterSize(25);

CircleShape circle5;

circle5.setRadius(55);

circle5.setFillColor(Color::Green);

circle5.setPosition(a5, b5);

circle5.setOutlineColor(Color::Black);

circle5.setOutlineThickness(2);

circle5.setPointCount(150);

int a7, b7;

a7 = 300;

b7 = 600;

sf::Text text7("7", font);

text7.setFillColor(sf::Color::Black);

text7.setPosition(a7 + 45, b7 + 40);

text7.setCharacterSize(25);

CircleShape circle7;

circle7.setRadius(55);

circle7.setFillColor(Color::Green);

circle7.setPosition(a7, b7);

circle7.setOutlineColor(Color::Black);

circle7.setOutlineThickness(2);

circle7.setPointCount(150);

int a6, b6;

a6 = 600;

b6 = 500;

sf::Text text6("6", font);

text6.setFillColor(sf::Color::Black);

text6.setPosition(a6 + 45, b6 + 40);

text6.setCharacterSize(25);

CircleShape circle6;

circle6.setRadius(55);

circle6.setFillColor(Color::Green);

circle6.setPosition(a6, b6);

circle6.setOutlineColor(Color::Black);

circle6.setOutlineThickness(2);

circle6.setPointCount(150);

int a4, b4;

a4 = 800;

b4 = 300;

sf::Text text4("4", font);

text4.setFillColor(sf::Color::Black);

text4.setPosition(a4 + 45, b4 + 40);

text4.setCharacterSize(25);

CircleShape circle4;

circle4.setRadius(55);

circle4.setFillColor(Color::Green);

circle4.setPosition(a4, b4);

circle4.setOutlineColor(Color::Black);

circle4.setOutlineThickness(2);

circle4.setPointCount(150);

int a3, b3;

a3 = 450;

b3 = 250;

sf::Text text3("3", font);

text3.setFillColor(sf::Color::Black);

text3.setPosition(a3 + 45, b3 + 40);

text3.setCharacterSize(25);

CircleShape circle3;

circle3.setRadius(55);

circle3.setFillColor(Color::Green);

circle3.setPosition(a3, b3);

circle3.setOutlineColor(Color::Black);

circle3.setOutlineThickness(2);

circle3.setPointCount(150);

//выводим полный путь

sf::Text text\_result("Full way:1->4->3->6->7->5->2->1", font);

text\_result.setFillColor(sf::Color::Black);

text\_result.setPosition(a4-50 , b4 -300);

text\_result.setCharacterSize(30);

//выводим кратчайшую длину пути

sf::Text text\_shortest("shortest path length=42", font);

text\_shortest.setFillColor(sf::Color::Black);

text\_shortest.setPosition(a4 - 50, b4 -200 );

text\_shortest.setCharacterSize(30);

//создаем массив точек, по которым будут рисоваться линии:

sf::VertexArray line1(sf::Lines, 16 /\*количество точек\*/);

//далее для каждой из точек задаем свою позицию:

line1[0].position = sf::Vector2f(a1+55, b1+60);

line1[1].position = sf::Vector2f(a2+55, b2+60);

//и т.д.

//далее для каждой точки указываем цвет(так можно создавать градиенты):

line1[0].color = sf::Color::Red;

line1[1].color = sf::Color::Red;

sf::VertexArray line2(sf::Lines, 16 /\*количество точек\*/);

line2[0].position = sf::Vector2f(a2 + 55, b2 + 60);

line2[1].position = sf::Vector2f(a5 + 55, b5 + 60);

line2[0].color = sf::Color::Black;

line2[1].color = sf::Color::Black;

sf::VertexArray line3(sf::Lines, 16 /\*количество точек\*/);

line3[0].position = sf::Vector2f(a5 + 55, b5 + 60);

line3[1].position = sf::Vector2f(a7 + 55, b7 + 60);

line3[0].color = sf::Color::Black;

line3[1].color = sf::Color::Black;

sf::VertexArray line4(sf::Lines, 16 /\*количество точек\*/);

line4[0].position = sf::Vector2f(a7 + 55, b7 + 60);

line4[1].position = sf::Vector2f(a6 + 55, b6 + 60);

line4[0].color = sf::Color::Black;

line4[1].color = sf::Color::Black;

sf::VertexArray line5(sf::Lines, 16 /\*количество точек\*/);

line5[0].position = sf::Vector2f(a6 + 55, b6 + 60);

line5[1].position = sf::Vector2f(a4 + 55, b4 + 60);

line5[0].color = sf::Color::Black;

line5[1].color = sf::Color::Black;

sf::VertexArray line6(sf::Lines, 16 /\*количество точек\*/);

line6[0].position = sf::Vector2f(a4 + 55, b4 + 60);

line6[1].position = sf::Vector2f(a3 + 55, b3 + 60);

line6[0].color = sf::Color::Black;

line6[1].color = sf::Color::Black;

sf::VertexArray line7(sf::Lines, 16 /\*количество точек\*/);

line7[0].position = sf::Vector2f(a3 + 55, b3 + 60);

line7[1].position = sf::Vector2f(a1 + 55, b1 + 60);

line7[0].color = sf::Color::Black;

line7[1].color = sf::Color::Black;

sf::VertexArray line8(sf::Lines, 16 /\*количество точек\*/);

line8[0].position = sf::Vector2f(a1 + 55, b1 + 60);

line8[1].position = sf::Vector2f(a2 + 55, b2 + 60);

line8[0].color = sf::Color::Black;

line8[1].color = sf::Color::Black;

sf::VertexArray line9(sf::Lines, 16 /\*количество точек\*/);

line9[0].position = sf::Vector2f(a1 + 55, b1 + 60);

line9[1].position = sf::Vector2f(a5 + 55, b5 + 60);

line9[0].color = sf::Color::Black;

line9[1].color = sf::Color::Black;

sf::VertexArray line10(sf::Lines, 16 /\*количество точек\*/);

line10[0].position = sf::Vector2f(a5 + 55, b5 + 60);

line10[1].position = sf::Vector2f(a6 + 55, b6 + 60);

line10[0].color = sf::Color::Black;

line10[1].color = sf::Color::Black;

sf::VertexArray line11(sf::Lines, 16 /\*количество точек\*/);

line11[0].position = sf::Vector2f(a5 + 55, b5 + 60);

line11[1].position = sf::Vector2f(a6 + 55, b6 + 60);

line11[0].color = sf::Color::Black;

line11[1].color = sf::Color::Black;

sf::VertexArray line12(sf::Lines, 16 /\*количество точек\*/);

line12[0].position = sf::Vector2f(a3 + 55, b3 + 60);

line12[1].position = sf::Vector2f(a6 + 55, b6 + 60);

line12[0].color = sf::Color::Black;

line12[1].color = sf::Color::Black;

sf::VertexArray line13(sf::Lines, 16 /\*количество точек\*/);

line13[0].position = sf::Vector2f(a1 + 55, b1 + 60);

line13[1].position = sf::Vector2f(a5 + 55, b5 + 60);

line13[0].color = sf::Color::Black;

line13[1].color = sf::Color::Black;

sf::VertexArray line14(sf::Lines, 16 /\*количество точек\*/);

line14[0].position = sf::Vector2f(a4 + 55, b4 + 60);

line14[1].position = sf::Vector2f(a1 + 55, b1 + 60);

line14[0].color = sf::Color::Black;

line14[1].color = sf::Color::Black;

sf::Text ves\_1\_2("9", font, 20);

ves\_1\_2.setStyle(sf::Text::Bold);

ves\_1\_2.setFillColor(sf::Color::Red);

ves\_1\_2.setPosition(a1-20, b1+70);

sf::Text ves\_1\_5("7", font, 20);

ves\_1\_5.setStyle(sf::Text::Bold);

ves\_1\_5.setFillColor(sf::Color::Red);

ves\_1\_5.setPosition(a2+120, b2);

sf::Text ves\_1\_3("5", font, 20);

ves\_1\_3.setStyle(sf::Text::Bold);

ves\_1\_3.setFillColor(sf::Color::Red);

ves\_1\_3.setPosition(a3-330, b3+46);

sf::Text ves\_1\_4("8", font, 20);

ves\_1\_4.setCharacterSize(30);

ves\_1\_4.setStyle(sf::Text::Bold);

ves\_1\_4.setFillColor(sf::Color::Red);

ves\_1\_4.setPosition(a3-315, b4+90);

sf::Text ves\_2\_1("10", font, 20);

ves\_2\_1.setStyle(sf::Text::Bold);

ves\_2\_1.setFillColor(sf::Color::Red);

ves\_2\_1.setPosition(a3-240, b4+90);

sf::Text ves\_2\_5("13", font, 20);

ves\_2\_5.setStyle(sf::Text::Bold);

ves\_2\_5.setFillColor(sf::Color::Red);

ves\_2\_5.setPosition(a3-140, b4-120);

sf::Text ves\_5\_2("5", font, 20);

ves\_5\_2.setStyle(sf::Text::Bold);

ves\_5\_2.setFillColor(sf::Color::Red);

ves\_5\_2.setPosition(280, 500);

sf::Text ves\_5\_1("6", font, 20);

ves\_5\_1.setStyle(sf::Text::Bold);

ves\_5\_1.setFillColor(sf::Color::Red);

ves\_5\_1.setPosition(500, 520);

sf::Text ves\_5\_7("10", font, 20);

ves\_5\_7.setStyle(sf::Text::Bold);

ves\_5\_7.setFillColor(sf::Color::Red);

ves\_5\_7.setPosition(210, 530);

sf::Text ves\_5\_6("7", font, 20);

ves\_5\_6.setStyle(sf::Text::Bold);

ves\_5\_6.setFillColor(sf::Color::Red);

ves\_5\_6.setPosition(300, 590);

sf::Text ves\_7\_5("9", font, 20);

ves\_7\_5.setStyle(sf::Text::Bold);

ves\_7\_5.setFillColor(sf::Color::Red);

ves\_7\_5.setPosition(430, 610);

sf::Text ves\_7\_6("5", font, 20);

ves\_7\_6.setStyle(sf::Text::Bold);

ves\_7\_6.setFillColor(sf::Color::Red);

ves\_7\_6.setPosition(580, 560);

sf::Text ves\_6\_3("3", font, 20);

ves\_6\_3.setStyle(sf::Text::Bold);

ves\_6\_3.setFillColor(sf::Color::Red);

ves\_6\_3.setPosition(615, 470);

sf::Text ves\_6\_5("5", font, 20);

ves\_6\_5.setStyle(sf::Text::Bold);

ves\_6\_5.setFillColor(sf::Color::Red);

ves\_6\_5.setPosition(590, 300);

sf::Text ves\_6\_4("4", font, 20);

ves\_6\_4.setStyle(sf::Text::Bold);

ves\_6\_4.setFillColor(sf::Color::Red);

ves\_6\_4.setPosition(540, 350);

sf::Text ves\_4\_6("4", font, 20);

ves\_4\_6.setStyle(sf::Text::Bold);

ves\_4\_6.setFillColor(sf::Color::Red);

ves\_4\_6.setPosition(760, 320);

sf::Text ves\_4\_3("11", font, 20);

ves\_4\_3.setStyle(sf::Text::Bold);

ves\_4\_3.setFillColor(sf::Color::Red);

ves\_4\_3.setPosition(460, 230);

sf::Text ves\_4\_1("9", font, 20);

ves\_4\_1.setStyle(sf::Text::Bold);

ves\_4\_1.setFillColor(sf::Color::Red);

ves\_4\_1.setPosition(400, 140);

sf::Text ves\_3\_1("13", font, 20);

ves\_3\_1.setStyle(sf::Text::Bold);

ves\_3\_1.setFillColor(sf::Color::Red);

ves\_3\_1.setPosition(a4-20, b4+85);

sf::Text ves\_3\_6("11", font, 20);

ves\_3\_6.setStyle(sf::Text::Bold);

ves\_3\_6.setFillColor(sf::Color::Red);

ves\_3\_6.setPosition(700, 470);

sf::Text ves\_1\_9("7", font, 20);

ves\_1\_9.setStyle(sf::Text::Bold);

ves\_1\_9.setFillColor(sf::Color::Red);

ves\_1\_9.setPosition(a1+130 , b1+77 );

sf::Text ves\_1\_10("5", font, 20);

ves\_1\_10.setStyle(sf::Text::Bold);

ves\_1\_10.setFillColor(sf::Color::Red);

ves\_1\_10.setPosition(a4-50, b4-13);

//Подсказка

sf::Text Pod("Press escape to quit", font, 20);

Pod.setStyle(sf::Text::Bold);

Pod.setFillColor(sf::Color::Black);

Pod.setPosition(450, 700);

Pod.setCharacterSize(30);

while (window.isOpen())

{

sf::Event event;

while (window.pollEvent(event))

{

if (event.type == sf::Event::Closed)

window.close();

if (event.type == sf::Event::KeyPressed)

{

// Получаем нажатую клавишу - выполняем соответствующее действие

if (event.key.code == sf::Keyboard::Escape)

{

window.close();

}

}

window.clear(sf::Color(200, 200, 200));

window.draw(sprite);

window.draw(line1);

window.draw(line2);

window.draw(line3);

window.draw(line4);

window.draw(line5);

window.draw(line6);

window.draw(line7);

window.draw(line8);

window.draw(line9);

window.draw(line11);

window.draw(line12);

window.draw(line13);

window.draw(line14);

window.draw(line10);

window.draw(circle2);

window.draw(text2);

window.draw(circle1);

window.draw(text1);

window.draw(circle5);

window.draw(text5);

window.draw(circle7);

window.draw(text7);

window.draw(circle6);

window.draw(text6);

window.draw(circle4);

window.draw(text4);

window.draw(circle3);

window.draw(text3);

window.draw(text\_result);

window.draw(text\_shortest);

window.draw(ves\_1\_2);

window.draw(ves\_1\_5);

window.draw(ves\_1\_3);

window.draw(ves\_1\_4);

window.draw(ves\_2\_1);

window.draw(ves\_2\_5);

window.draw(ves\_5\_2);

window.draw(ves\_5\_1);

window.draw(ves\_5\_7);

window.draw(ves\_5\_6);

window.draw(ves\_7\_5);

window.draw(ves\_7\_6);

window.draw(ves\_6\_3);

window.draw(ves\_6\_5);

window.draw(ves\_6\_4);

window.draw(ves\_4\_6);

window.draw(ves\_4\_3);

window.draw(ves\_4\_1);

window.draw(ves\_3\_1);

window.draw(ves\_3\_6);

window.draw(ves\_1\_9);

window.draw(ves\_1\_10);

window.draw(Pod);

window.display();

}

}