Образец оформления отчёта по Лабораторной работе.

|  |  |
| --- | --- |
| **К Г Э У** | МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

**Кафедра Информатики и информационных управляющих систем**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**ПОСТРОЕНИЕ ПРОГРАММ С АЛГОРИТМАМИ ВЕТВЛЕНИЯ**

|  |  |
| --- | --- |
| **Исполнитель:** | Соловьёв Леонид |
| **Группа:** | ПИ-1-22 |
|  |  |
|  |  |

Казань – 2022

1.1  
  
Код  
#include <iostream>

#include <cmath>

#include <ctime>

const int m = rand()%10 + 10;

const int n = rand()%10;

const float mn = (float) m / float(n);

const float nm = (float) n / float(m);

double spec\_var(double t);

void xLowerThanPhi(float x, double phi);

void xEqualPhi(float x, double phi);

void xGreaterThanPhi(double phi);

int main() {

    srand(time(0));

    for (int i = 0; i < 2; i++) {

        double phi = spec\_var(rand()%1000);

        float x = rand()%100;

        std::cout << "phi = " << phi << std::endl;

        std::cout << "x = " << x << std::endl;

        if (x < phi) {

            xLowerThanPhi(x, phi);

        } else if (x = phi) {

            xEqualPhi(x, phi);

        } else {

            xGreaterThanPhi(phi);

        }

        std::cout << '\n';

    }

    return 0;

}

double spec\_var(double t) {

    return std::exp(std::sin(t)) - std::exp(std::cos(t));

}

void xLowerThanPhi(float x, double phi) {

    float zetta = std::pow(std::sin(mn), std::pow(x, 2) / (x + phi));

    std::cout << "z = " << zetta;

}

void xEqualPhi(float x, double phi) {

    float zetta = std::pow(phi, 2) \* std::pow(x, n/4.) / (std::pow(phi, nm) \* x);

    std::cout << "z = " << zetta;

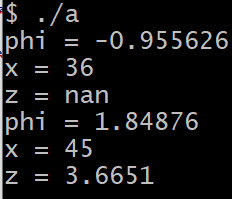
}

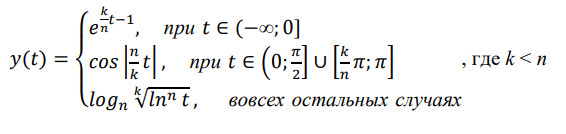
void xGreaterThanPhi(double phi) {

    float zetta = std::pow((std::log(std::abs(phi)) / std::log(mn)), nm);

    std::cout << "z = " << zetta;

}

Решение  


1.2  
  
Код  
#include <iostream>

#include <cmath>

#include <ctime>

const int n = rand()%10 + 10;

const int k = rand()%10;

const float kn = (float) k / float(n);

const float nk = (float) n / float(k);

const double pi = std::acos(-1.);

void uniq(float t); void tLessThan0(float t); void tGreaterThanPi(float t);

int main() {

    for (int i = 0; i < 2; i++) {

        srand(time(0) + i);

        float t = (float) (rand()%1000 + 1) / (rand()%1000 + 1);

        std::cout << "t = " << t;

        if (t <= 0) {

            tLessThan0(t);

        } else if (t >= pi) {

            uniq(t);

        } else {

            tGreaterThanPi(t);

        }

        std::cout << '\n';

    }

}

void uniq(float t) {

    std::cout << "\ny = " << std::cos(std::abs(nk\*t));

}

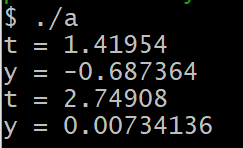
void tLessThan0(float t) {

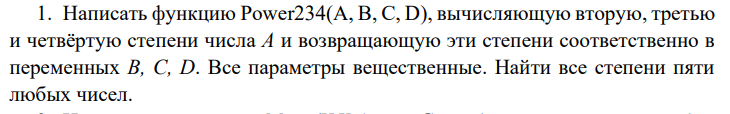
    std::cout << "\ny = " << exp(kn\*t - 1);

}

void tGreaterThanPi(float t) {

    std::cout << "\ny = " << std::log(std::pow(std::log(t), nk)) / std::log(n);

}  
Решение  


2.1  
  
Код  
#include <iostream>

#include <cmath>

void Power234(int i, int &b, int &c, int &d);

int main() {

    int b, c, d;

    for (int i = 0; i < 10; i++) {

        Power234(i, b, c, d);

        std::cout << i << "  second grade: " << b << "  third grade: " << c

        << "  fourth grade: " << d << '\n';

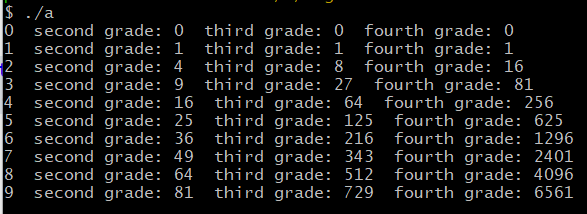
    }

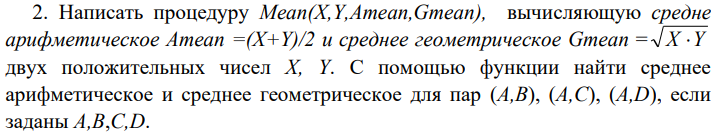
    return 0;

}

void Power234(int i, int &b, int &c, int &d) {

    b = i \* i; c = i \* i \* i; d = i \* i \* i \* i;

}  
Решение  


2.2  
  
Код  
#include <iostream>

#include <cmath>

void Mean(float X, float Y, float Amean, float Gmean);

int main() {

    float A = 0, B = 0, C = 0, D = 0, Amean = 0, Gmean = 0;

    std::cout << "Input A: "; std::cin >> A;

    std::cout << "Input B: "; std::cin >> B;

    Mean(A, B, Amean, Gmean);

    std::cout << "\nInput C: "; std::cin >> C;

    Mean(A, C, Amean, Gmean);

    std::cout << "\nInput D: "; std::cin >> D;

    Mean(A, D, Amean, Gmean);

    return 0;

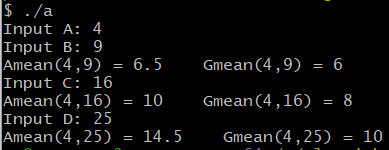
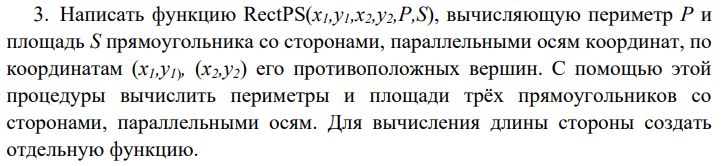
}

void Mean(float X, float Y, float Amean, float Gmean) {

    Amean = (X + Y) / 2.;

    Gmean = std::pow(X \* Y, 0.5);

    std::cout << "Amean(" << X << ',' << Y << ") = " << Amean << "    Gmean(" << X << ',' << Y << ") = " << Gmean;

}  
Решение  
  
  
  
2.3  
  
Код  
#include <iostream>

#include <cmath>

void RectPS( float x1, float y1, float x2, float y2, float P, float S);

int main() {

    float x1 = 0, x2 = 0, y1 = 0, y2 = 0, P = 0, S = 0;

    char s = 'y';

    while (s == 'y') {

        std::cout << "Enter x1: "; std::cin >> x1;

        std::cout << "Enter y1: "; std::cin >> y1;

        std::cout << "Enter x2: "; std::cin >> x2;

        std::cout << "Enter y2: "; std::cin >> y2;

        RectPS(x1, y1, x2, y2, P, S);

        std::cout << "\nDo you wanna calculate perimeter and square of next rectangle?\nType y(yes) or n(n): ";

        std::cin >> s;

        std::cout << '\n';

    }

    std::cout << "Thanks for using this program!";

    return 0;

}

void RectPS( float x1, float y1, float x2, float y2, float P, float S) {

    float a = std::abs(x2 - x1), b = std::abs(y2 - y1);

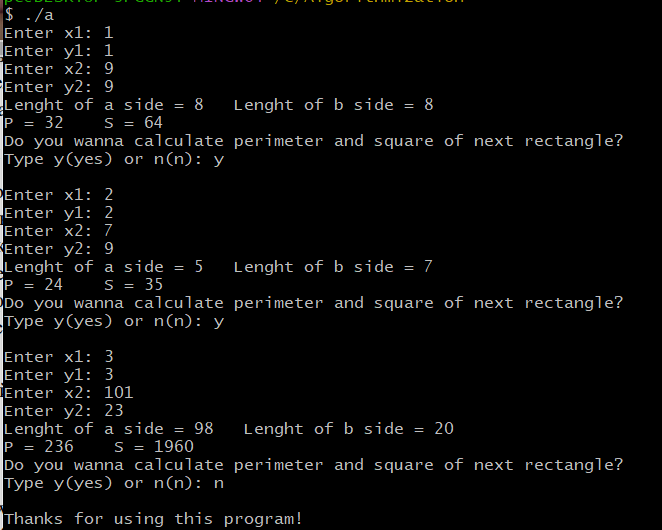
    std::cout << "Lenght of a side = " << a << "   Lenght of b side = " << b << '\n';

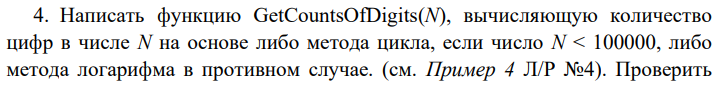
    P = (a + b) \* 2;

    S = a \* b;

    std::cout << "P = " << P << "    S = " << S;

}

Решение  


2.4  
  
  
Код  
#include <iostream>

#include <cmath>

void GetCountOfDigits(unsigned N);

int main() {

    srand(time(0));

    unsigned N = 0;

    char flag = 'y';

    int i = 0;

    while (i < 10) {

        N = (rand() \* rand()) % (rand() \* rand());

        GetCountOfDigits(N);

        i++;

    }

    return 0;

}

void GetCountOfDigits(unsigned N) {

    unsigned num = N;

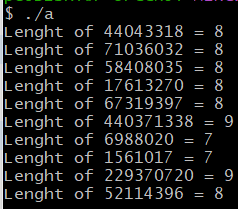
    for (int i = 0; i < 10000; i++) {

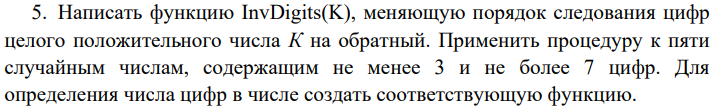
        if (N < std::pow(10, i + 1)) {

            std::cout << "Lenght of " << num << " = " << i+1 << std::endl;

            break;

        }

    }  
  
Решение  


2.5  
  
Код  
#include <iostream>

#include <cmath>

#include <ctime>

void InvDigits(unsigned K);

int main() {

    srand(time(0));

    unsigned K = 0;

    char flag = 'y';

    int i = 0;

    while (i < 5) {

        K = (rand() \* (rand()%100))%9999900 + 100;

        InvDigits(K);

        i++;

    }

    return 0;

}

void InvDigits(unsigned K) {

    unsigned num = K;

    (num % 10 == 0) ? num++ : num;

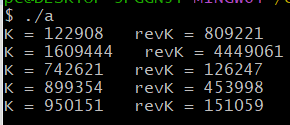
    std::cout << "K = " << num << "   revK = ";

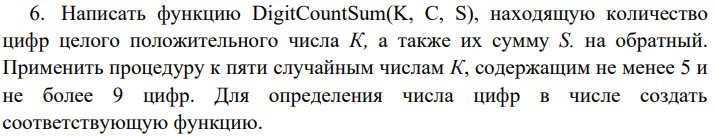
    while (num > 0) {

        std::cout << num % 10;

        num /= 10;

    }

    std::cout << '\n';  
}  
Решение  


2.6  
  
Код  
#include <iostream>

#include <cmath>

#include <ctime>

void DigitCountSum(unsigned N);

int main() {

    srand(time(0));

    unsigned N = 0;

    int i = 0;

    while (i < 10) {

        N = (rand() \* rand()) % (rand() \* rand());

        DigitCountSum(N);

        i++;

    }

    return 0;

}

void DigitCountSum(unsigned N) {

    unsigned num = N;

    unsigned sum = 0;

    for (int i = 0; i < 10000; i++) {

        if (N < std::pow(10, i + 1)) {

            std::cout << "Lenght of " << num << " = " << i+1 << "   Sum of " << num << " = ";

            break;

        }

    }

    while (num > 0) {

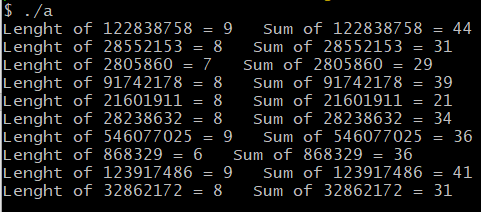
        sum += num % 10;

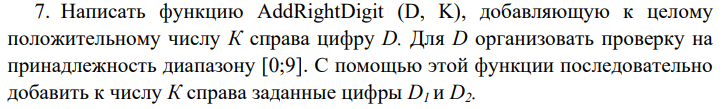
        num /= 10;

    }

    std::cout << sum << std::endl;

}

Решение  


2.7  
  
Код  
#include <iostream>

#include <cmath>

void AddRightDigit(unsigned D, unsigned &K);

int main() {

    unsigned K = 0, D\_1 = 0, D\_2 = 0;

    std::cout << "Enter K: "; std::cin >> K;

    std::cout << "Enter D1: "; std::cin >> D\_1;

    AddRightDigit(D\_1, K);

    std::cout << "Enter D2: "; std::cin >> D\_2;

    AddRightDigit(D\_2, K);

    return 0;

}

void AddRightDigit(unsigned D, unsigned &K) {

    if (0 <= D && D < 10) {

        std::cout << "Old K = " << K << std::endl;

        K = K \* 10 + D;

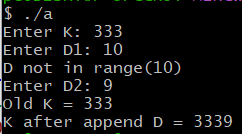
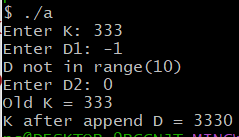
        std::cout << "K after append D = " << K;

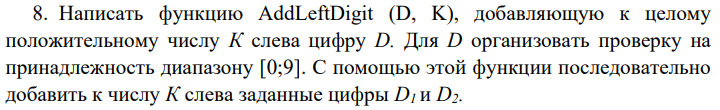
    } else {

        std::cout << "D not in range(10)\n";

    }

}

Решение  
 

2.8  
  
Код  
#include <iostream>

#include <cmath>

void AddRightDigit(unsigned D, unsigned &K);

int Lenght(double K);

int main() {

    unsigned K = 0; unsigned D\_1 = 0, D\_2 = 0;

    std::cout << "Enter K: "; std::cin >> K;

    std::cout << "Enter D1: "; std::cin >> D\_1;

    AddRightDigit(D\_1, K);

    std::cout << "Enter D2: "; std::cin >> D\_2;

    AddRightDigit(D\_2, K);

    return 0;

}

void AddRightDigit(unsigned D, unsigned &K) {

    if (0 <= D && D < 10) {

        std::cout << "Old K = " << K << std::endl;

        K = ((K / std::pow(10, Lenght((double) K))) + D) \* std::pow(10, Lenght((double) K));

        std::cout << "K after append D = " << K << std::endl;

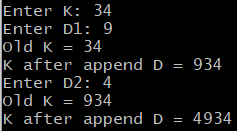
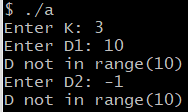
    } else {

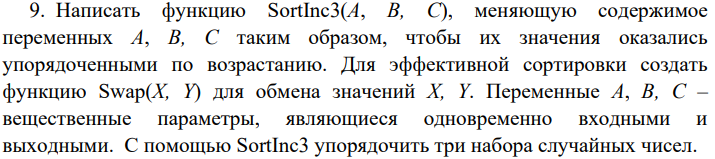
        std::cout << "D not in range(10)\n";

    }

}

int Lenght(double K) {

    return (K == 0) ? 1 : (int) std::ceil(log10(std::abs(K) + 0.5));  
}  
Решение  
 

2.9  


Код  
#include <iostream>

#include <cmath>

void SortInc3(int &A, int &B, int &C);

void Swap(int &A, int &B);

int main() {

    int A = 0, B = 0, C = 0;

    std::cout << "Enter A: "; std::cin >> A;

    std::cout << "Enter B: "; std::cin >> B;

    std::cout << "Enter C: "; std::cin >> C;

    std::cout << "Before sort: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    SortInc3(A, B, C);

    std::cout << "After sort: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    return 0;

}

void Swap(int &A, int &B) {

    int temporary = B;

    B = A;

    A = temporary;

}

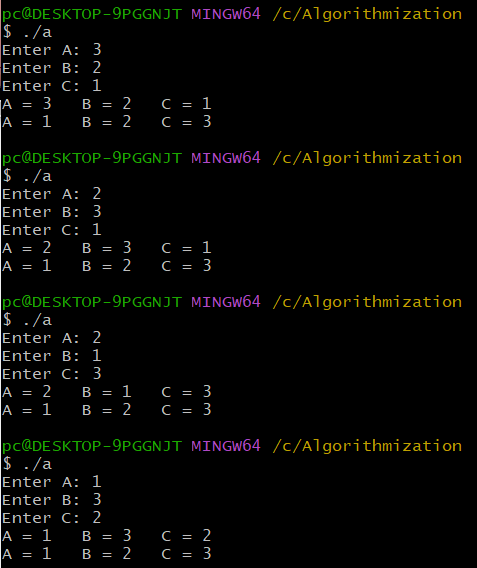
void SortInc3(int &A, int &B, int &C) {

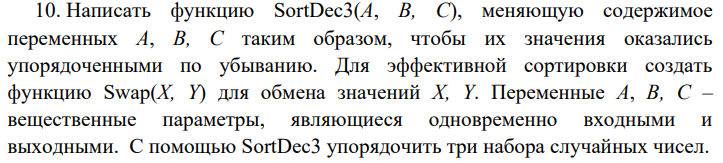
    if (A > B) Swap(A, B);

    if (B > C) Swap(B, C);

    if (B < A) Swap(B, A);

}

Решение  


2.10  
  
Код

#include <iostream>

#include <cmath>

void SortDec3(int &A, int &B, int &C);

void Swap(int &A, int &B);

int main() {

    int A = 0, B = 0, C = 0;

    std::cout << "Enter A: "; std::cin >> A;

    std::cout << "Enter B: "; std::cin >> B;

    std::cout << "Enter C: "; std::cin >> C;

    std::cout << "Before sort: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    SortInc3(A, B, C);

    std::cout << "After sort: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    return 0;

}

void Swap(int &A, int &B) {

    int temporary = A;

    A = B;

    B = temporary;

}

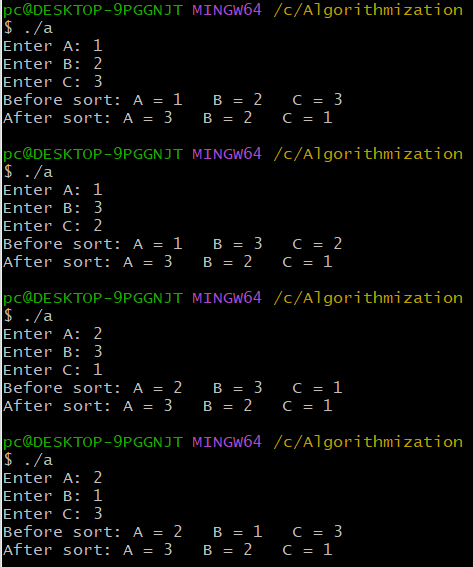
void SortDec3(int &A, int &B, int &C) {

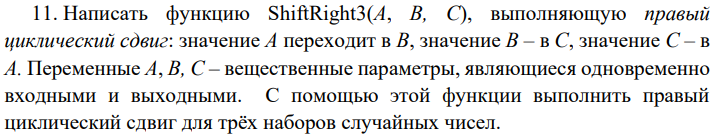
    if (A < B) Swap(A, B);

    if (B < C) Swap(B, C);

    if (B > A) Swap(B, A);

}

Решение  


2.11  
  
Код  
#include <iostream>

#include <cmath>

void ShiftRight3(int &A, int &B, int &C);

void Swap(int &A, int &B);

int main() {

    int A = 0, B = 0, C = 0;

    std::cout << "Enter A: "; std::cin >> A;

    std::cout << "Enter B: "; std::cin >> B;

    std::cout << "Enter C: "; std::cin >> C;

    std::cout << "Before shift: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    ShiftRight3(A, B, C);

    std::cout << "After shift: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    return 0;

}

void Swap(int &A, int &B) {

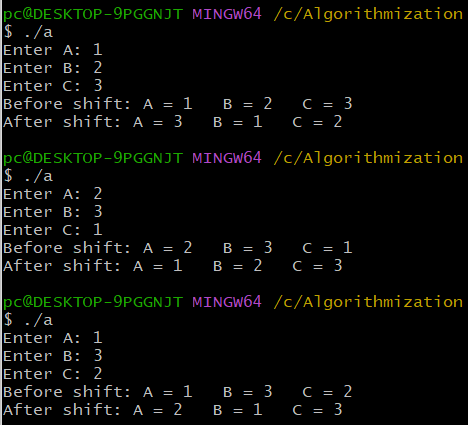
    int temporary = B;

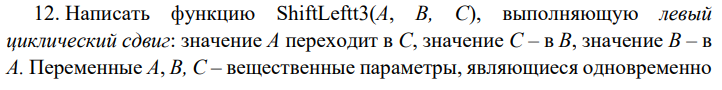
    B = A;

    A = temporary;

}

void ShiftRight3(int &A, int &B, int &C) {

    Swap(B, C); Swap(A, B);  
}  
Решение  


2.12  
  
  
Код  
#include <iostream>

#include <cmath>

void ShiftLeft3(int &A, int &B, int &C);

void Swap(int &A, int &B);

int main() {

    int A = 0, B = 0, C = 0;

    std::cout << "Enter A: "; std::cin >> A;

    std::cout << "Enter B: "; std::cin >> B;

    std::cout << "Enter C: "; std::cin >> C;

    std::cout << "Before shift: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    ShiftLeft3(A, B, C);

    std::cout << "After shift: A = " << A << "   B = " << B << "   C = " << C << std::endl;

    return 0;

}

void Swap(int &A, int &B) {

    int temporary = B;

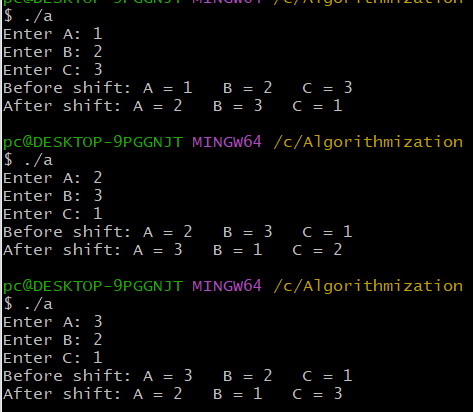
    B = A;

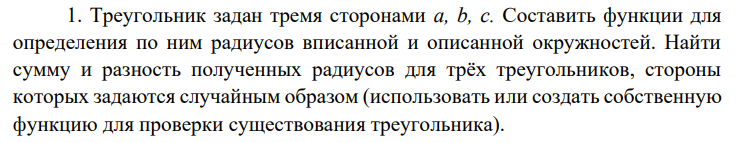
    A = temporary;

}

void ShiftLeft3(int &A, int &B, int &C) {

    Swap(A, B); Swap(B, C);

}  
  
Решение  


ДЗ1  


Код

#include <iostream>

#include <cmath>

#include <ctime>

void randSides(unsigned &a, unsigned &b, unsigned &c);

bool existance(unsigned a, unsigned b, unsigned c);

bool triangleExist(unsigned a, unsigned b, unsigned c);

int main() {

    for (int i = 0; i < 3; i++) {

        srand(time(0) + i \* i);

        unsigned a, b, c;

        randSides(a, b, c);

        std::cout << "\nRandom sides:\n\ta = " << a << "\n\tb = " << b << "\n\tc = " << c << std::endl;

        if (triangleExist(a, b, c)) {

            std::cout << "Triangle exists!\n";

            float p = (a + b + c) / 2.;

            std::cout << "\tp = " << p << std::endl;

            float S = std::sqrt(p \* (p-a) \* (p-b) \* (p - c));

            std::cout << "\tS = " << S << std::endl;

            float R = (a \* b \* c) / (4 \* S);

            std::cout << "\tR = " << R << std::endl;

            float r = std::sqrt(S / (p\*p));

            std::cout << "\tr = " << r << std::endl;

            std::cout << "\tR - r = " << R - r << std::endl;

            std::cout << "\tR + r = " << R + r << std::endl;

        } else {

            std::cout << "Triangle does not exist!\n";

        }

    }

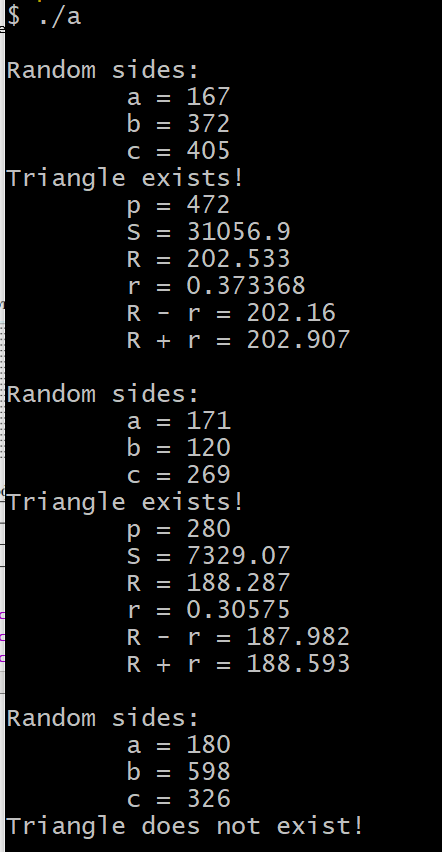
    return 0;

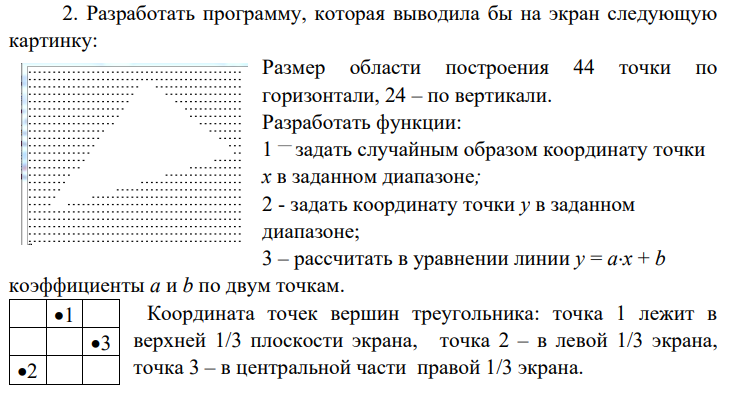
}

void randSides(unsigned &a, unsigned &b, unsigned &c) {a = rand()%1000;b = rand()%1000;c = rand()%1000;}

bool triangleExist(unsigned a, unsigned b, unsigned c) { return ((a < b + c) && (b < a + c) && (c < a + b));}

Решение



ДЗ2  


Код  
#include <iostream>

#include <cmath>

#include <ctime>

void Xcoordinates(unsigned &x\_1, unsigned &x\_2, unsigned &x\_3);

void Ycoordinates(unsigned &y\_1, unsigned &y\_2, unsigned &y\_3);

void coutCoordinates(unsigned &x\_1, unsigned &x\_2, unsigned &x\_3, unsigned &y\_1, unsigned &y\_2, unsigned &y\_3);

void abparameters(unsigned x\_1, unsigned y\_1, unsigned x\_2, unsigned y\_2, double &a, double &b);

void Triangle(unsigned x\_1, unsigned x\_2, unsigned x\_3, unsigned y\_1, unsigned y\_2, unsigned y\_3, double a\_12, double b\_12, double a\_23, double b\_23, double a\_31, double b\_31);

void coutTriangle();

char mass[45][24];

int main() {

    srand(time(0));

    unsigned x\_1, y\_1, x\_2, y\_2, x\_3, y\_3;

    double a\_12, b\_12, a\_23, b\_23, a\_31, b\_31;

    Xcoordinates(x\_1, x\_2, x\_3);

    Ycoordinates(y\_1, y\_2, y\_3);

    coutCoordinates(x\_1, x\_2, x\_3, y\_1, y\_2, y\_3);

    abparameters(x\_1, y\_1, x\_2, y\_2, a\_12, b\_12); //ab from 1 to 2

    abparameters(x\_2, y\_2, x\_3, y\_3, a\_23, b\_23); //ab from 2 to 3

    abparameters(x\_3, y\_3, x\_1, y\_1,a\_31, b\_31); //ab from 3 to 1

    Triangle(x\_1, x\_2, x\_3, y\_1, y\_2, y\_3, a\_12, b\_12, a\_23, b\_23, a\_31, b\_31);

    coutTriangle();

}

void Xcoordinates(unsigned &x\_1, unsigned &x\_2, unsigned &x\_3) {

    x\_1 = rand()%15 + 15;

    x\_2 = rand()%15;

    x\_3 = rand()%15 + 30;}

void Ycoordinates(unsigned &y\_1, unsigned &y\_2, unsigned &y\_3) {

    y\_1 = rand()%8;

    y\_2 = rand()%8 + 16;

    y\_3 = rand()%8 + 8;}

void coutCoordinates(unsigned &x\_1, unsigned &x\_2, unsigned &x\_3, unsigned &y\_1, unsigned &y\_2, unsigned &y\_3) {

    std::cout << "Сoordinates of first dot: (" << x\_1 << ',' << y\_1 << ")" << std::endl;

    std::cout << "Сoordinates of second dot: (" << x\_2 << ',' << y\_2 << ")" << std::endl;

    std::cout << "Сoordinates of third dot: (" << x\_3 << ',' << y\_3 << ")" << std::endl;

}

void abparameters(unsigned x\_1, unsigned y\_1, unsigned x\_2, unsigned y\_2, double &a, double &b) {

    double x1 = (double) x\_1, y1 = (double) y\_1, x2 = (double) x\_2, y2 = (double) y\_2;

    a = (y2-y1) / (x2-x1);

    b = (x1\*y2 - x2\*y1) / (x1-x2);

    // (y1-y2)x + (x1y2 - x2y1) = (x1-x2)y

    // y = ((y1-y2)/(x1-x2))x + ((x1y2 - x2y1) / (x1-x2))

}

void Triangle(unsigned x\_1, unsigned x\_2, unsigned x\_3, unsigned y\_1, unsigned y\_2, unsigned y\_3, double a\_12, double b\_12, double a\_23, double b\_23, double a\_31, double b\_31) {

    for (int k = 0; k <= 23; k++) {

        for (int i = 0; i <= 44; i++) {

            if ((k >= (int) (i \* a\_12 + b\_12)) && (k <= (int) (i \* a\_23 + b\_23)) && (k >= (int) (i \* a\_31 + b\_31))) {

                mass[i][k] = ' ';

            } else {

                mass[i][k] = '.';

            }

        }

    }

}

void coutTriangle() {

    for (int k = 0; k <= 23; k++) {

        for (int i = 0; i <= 44; i++) {

            std::cout << mass[i][k];

        }

        std::cout << '\n';

    }

}

Решение   
