**Московский авиационный институт (национальный исследовательский университет)**

**Институт №3.**

«Системы управления, информатика и электроэнергетика»

**Кафедра №304**

«Автоматизированные системы обработки информации и управления»

Отчет по тестам

по учебной дисциплине

«Численные методы»

Группа М30-207Б

Выполнил:

Гордеев Н.М.

Принял:

Яганов В.М.

**МОСКВА 2020**





Части кода программы

double Fx20(double x) { return 3 \* x + 4 \* x \* x \* x - 12 \* x \* x - 5; }

cout << "\t\t\t№ 1 " << endl;

cout << "Ответ: x = " << ПоловинноеДеление(0, 3, Fx20, 0.001) << endl;

double ПоловинноеДеление(double a1, double b1, double (\*F)(double), double Eps) {

const int n = 5;

int i = 0;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " a " };

char s2[15] = { " b " };

char s3[15] = { " x " };

char s4[15] = { " f(x) " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

s[4] = s4;

N\_tabl(n, s);

N[0] = i;

N[1] = a1;

N[2] = b1;

N[3] = a1;

N[4] = F(a1);

double a = a1;

double b = b1;

double x = a;

if (F(a) < F(b)) { a = b; b = x; x = a; }

while (abs(F(x)) > Eps)

{

C\_tabl(n, N);

i++;

if (F(x) < 0)

b = x;

else

a = x;

x = (a + b) / 2;

N[0] = i;

N[1] = a;

N[2] = b;

N[3] = x;

N[4] = F(x);

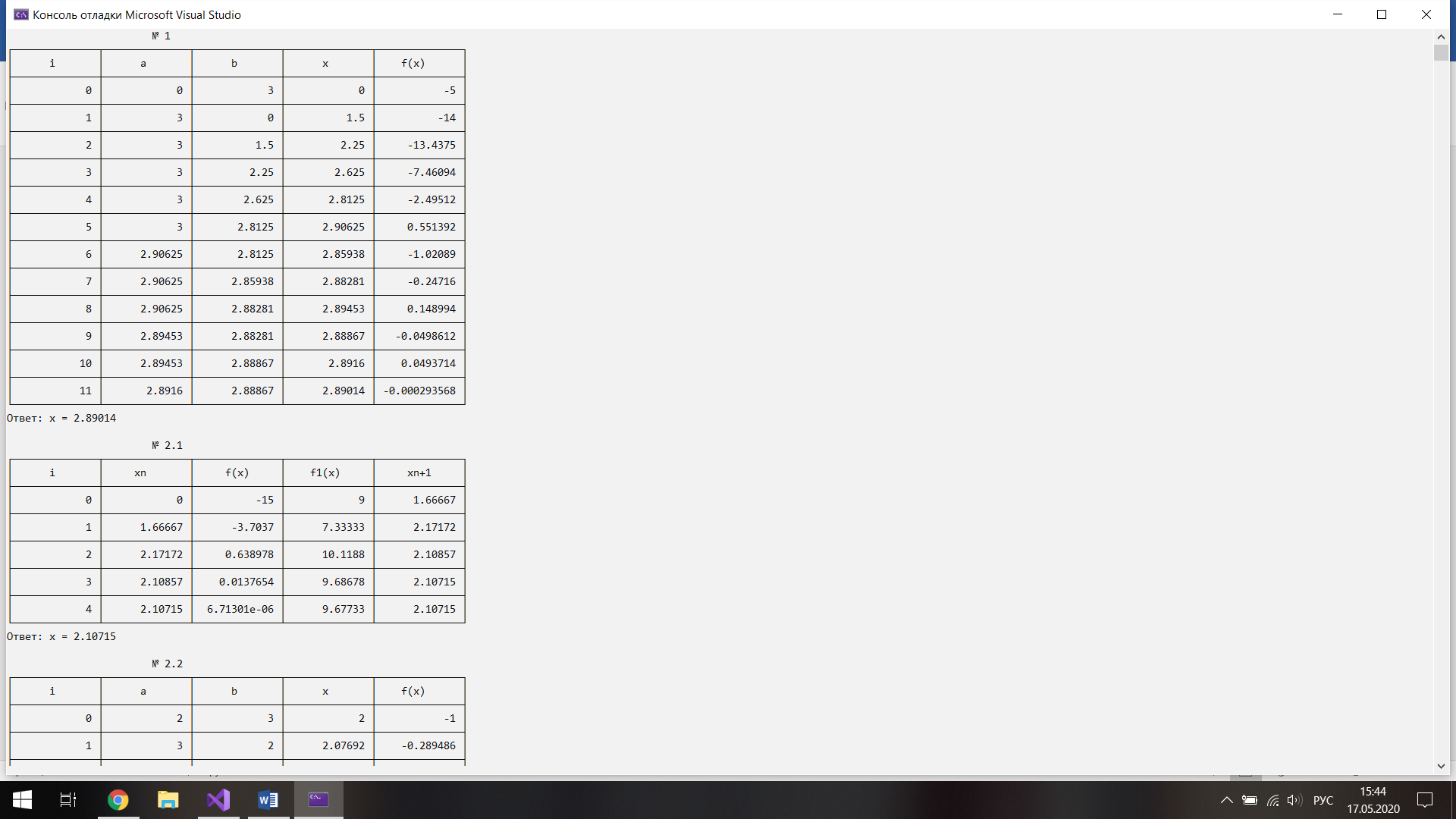
}

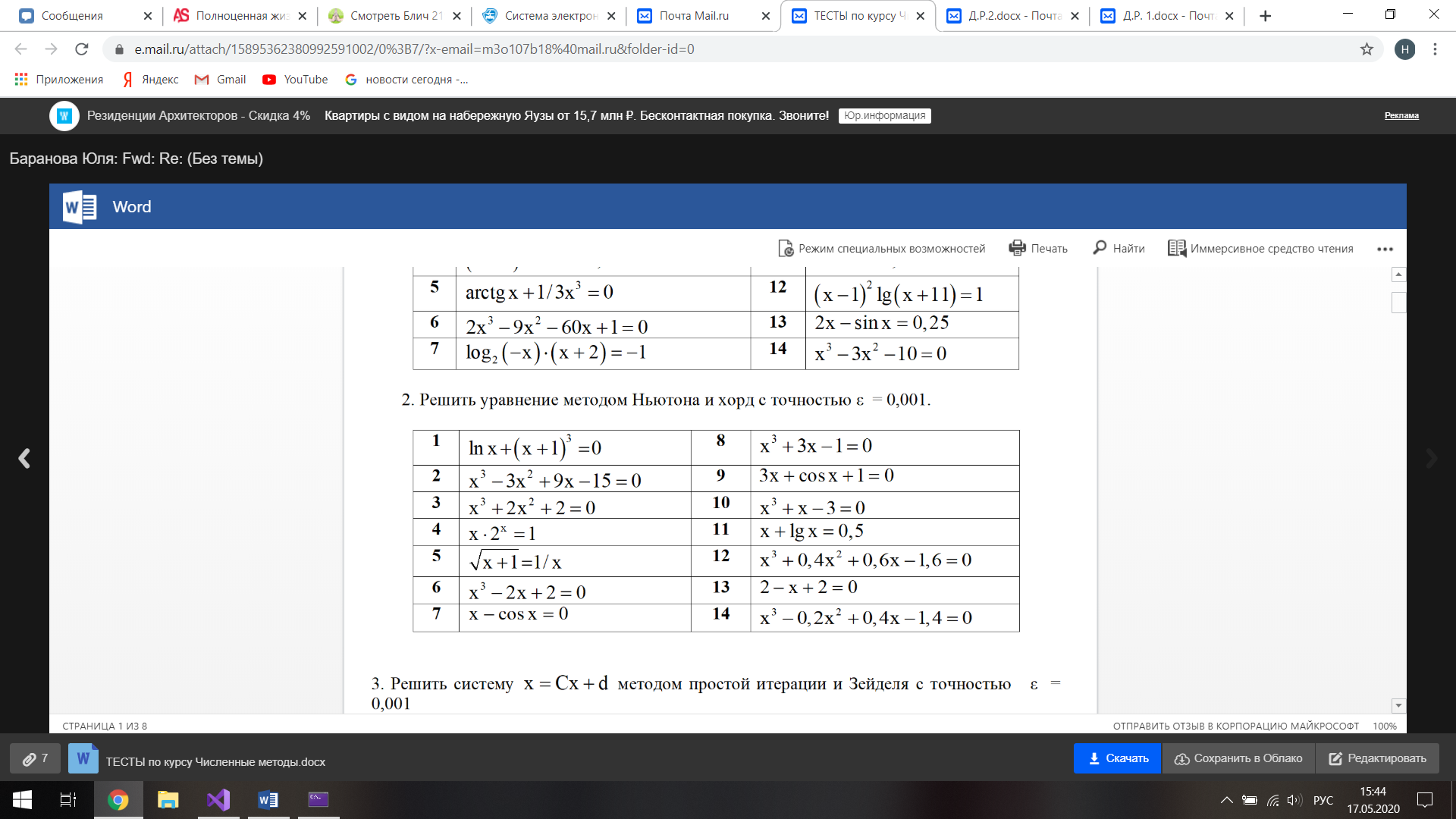
K\_tabl(n, N);

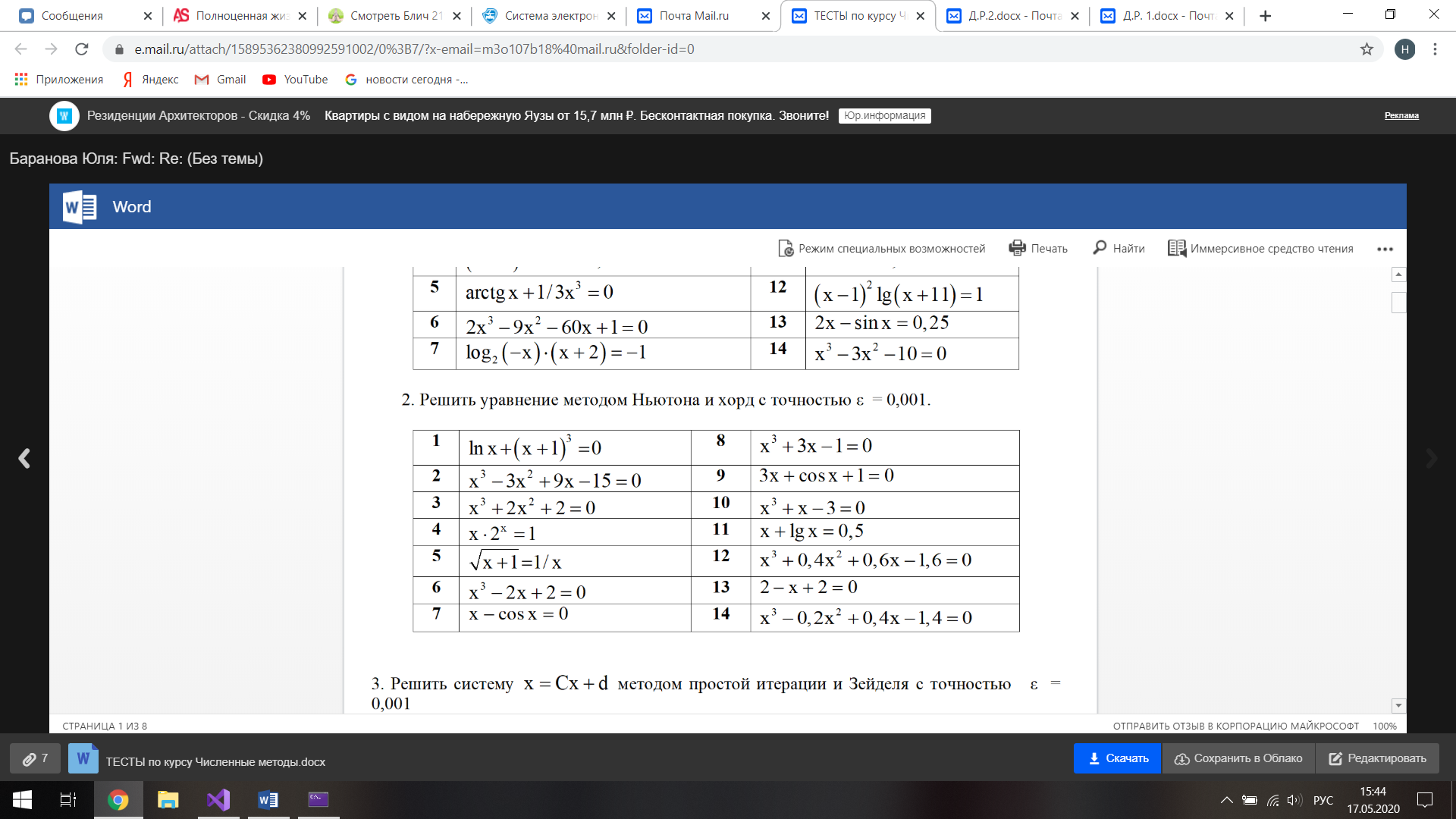
return x;

}

Результат работы программы







Части кода программы

double Fx21(double x) { return x \* x \* x - 3 \* x \* x + 9 \* x - 15; }

double Fx31(double x) { return 3\*x\*x - 6\*x + 9; }

cout << "\n\t\t\t№ 2.1 " << endl;

cout << "Ответ: x = " << Ньютон(0, Fx21, Fx31, 0.001) << endl;

cout << "\n\t\t\t№ 2.2 " << endl;

cout << "Ответ: x = " << МетодХорд(2, 3, Fx21, 0.001) << endl;

double Ньютон(double x1, double (\*F)(double), double (\*F1)(double), double Eps) {

const int n = 5;

int i = 0;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " xn " };

char s2[15] = { " f(x) " };

char s3[15] = { " f1(x) " };

char s4[15] = { " xn+1 " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

s[4] = s4;

N\_tabl(n, s);

N[0] = i;

N[1] = x1;

N[2] = F(x1);

N[3] = F1(x1);

N[4] = x1 - F(x1) / F1(x1);

double x = x1;

while (abs(F(x)) > Eps)

{

C\_tabl(n, N);

i++;

x = x - F(x) / F1(x);

N[0] = i;

N[1] = x;

N[2] = F(x);

N[3] = F1(x);

N[4] = x - F(x) / F1(x);

}

K\_tabl(n, N);

return x;

}

double МетодХорд(double a1, double b1, double (\*F)(double), double Eps) {

const int n = 5;

int i = 0;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " a " };

char s2[15] = { " b " };

char s3[15] = { " x " };

char s4[15] = { " f(x) " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

s[4] = s4;

N\_tabl(n, s);

N[0] = i;

N[1] = a1;

N[2] = b1;

N[3] = a1;

N[4] = F(a1);

double a = a1;

double b = b1;

double x = b;

bool t = 0;

if (a < b) { b = a; a = x; x = b; }

if (F(a) > 0) t = 1;

while (abs(F(x)) > Eps)

{

C\_tabl(n, N);

i++;

if(t)

x = x - F(x) / (F(x) - F(a)) \* (x - a);

else

x = x - F(x) / (F(b) - F(x)) \* (b - x);

N[0] = i;

N[1] = a;

N[2] = b;

N[3] = x;

N[4] = F(x);

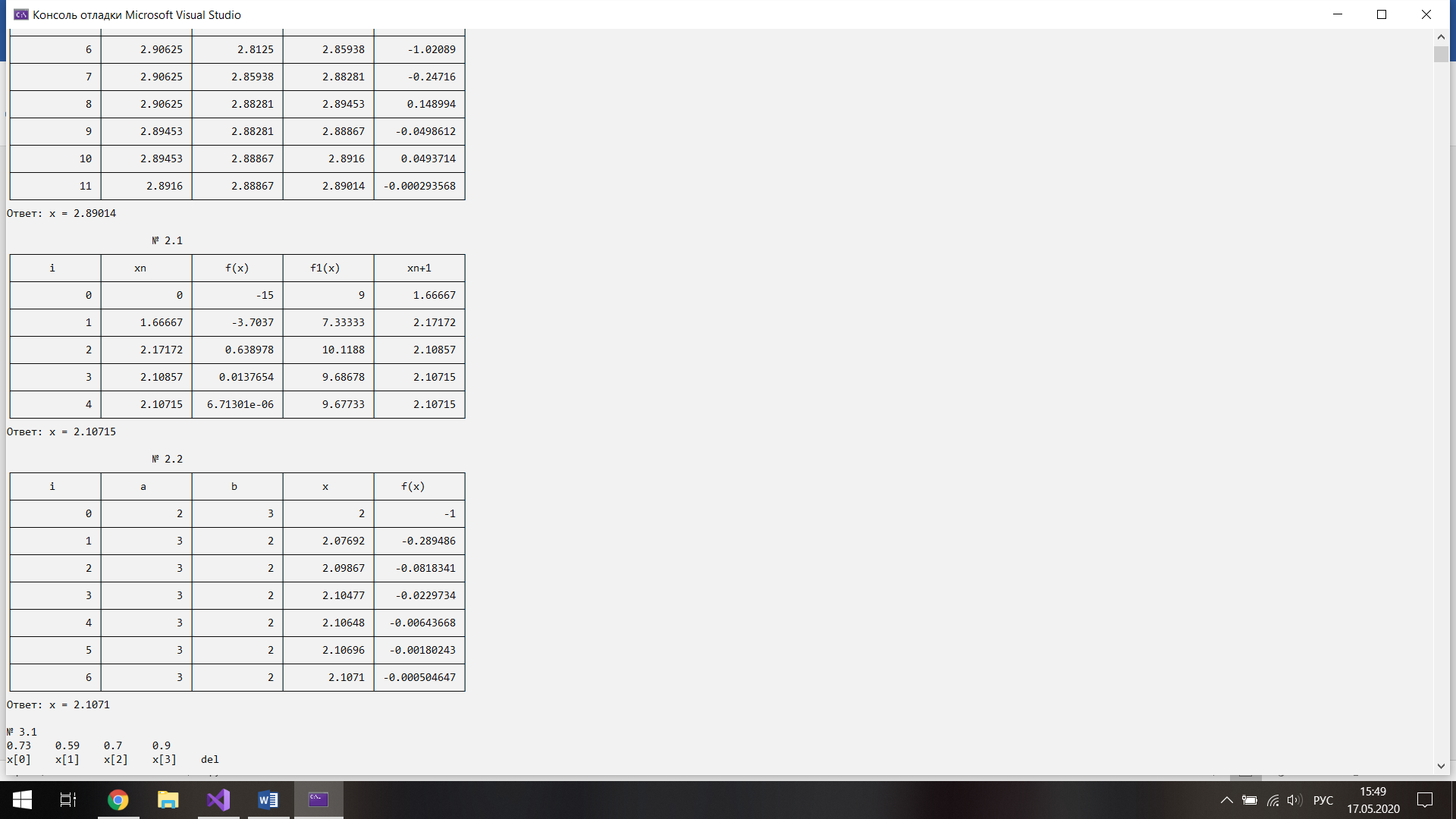
}

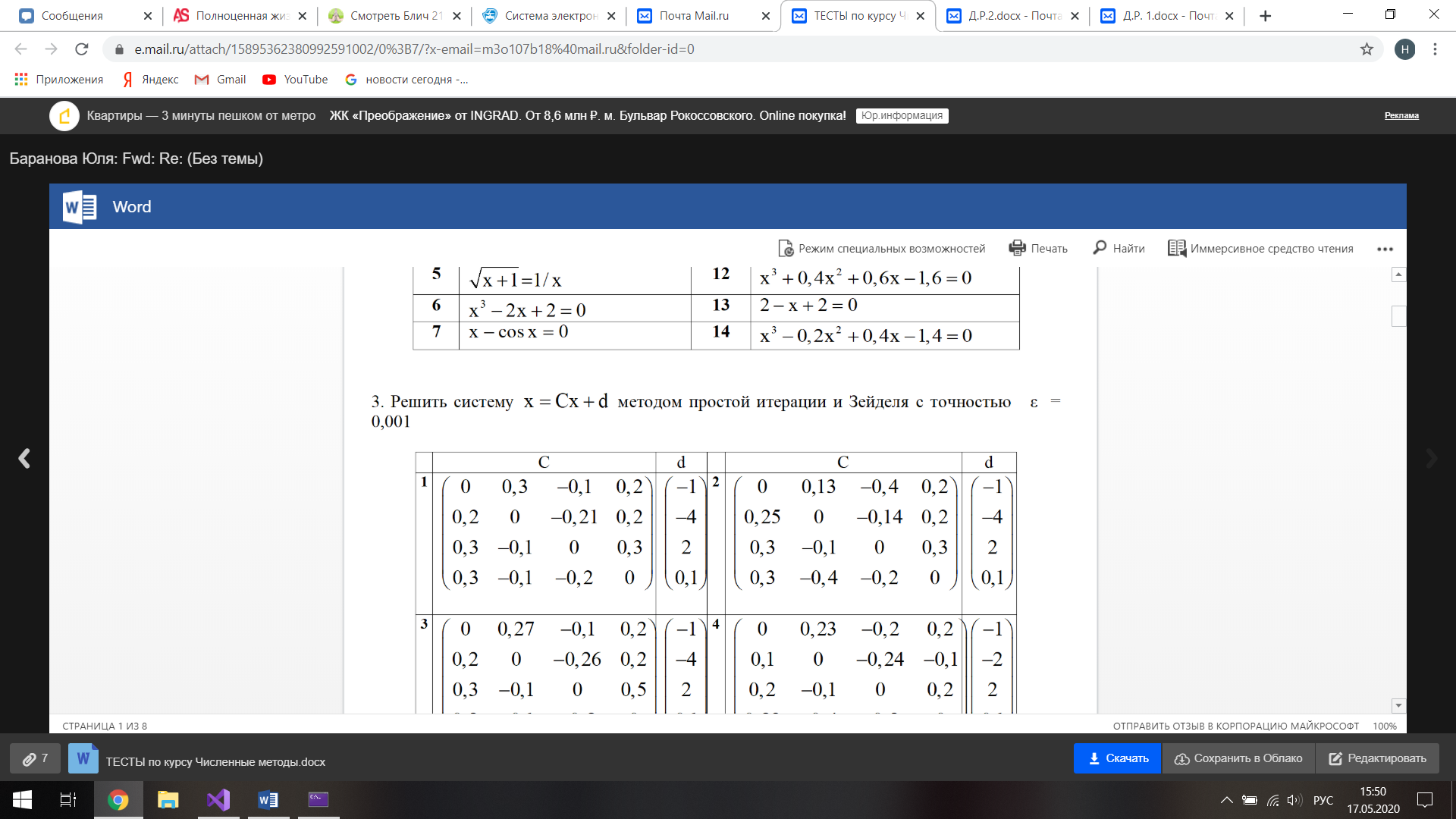
K\_tabl(n, N);

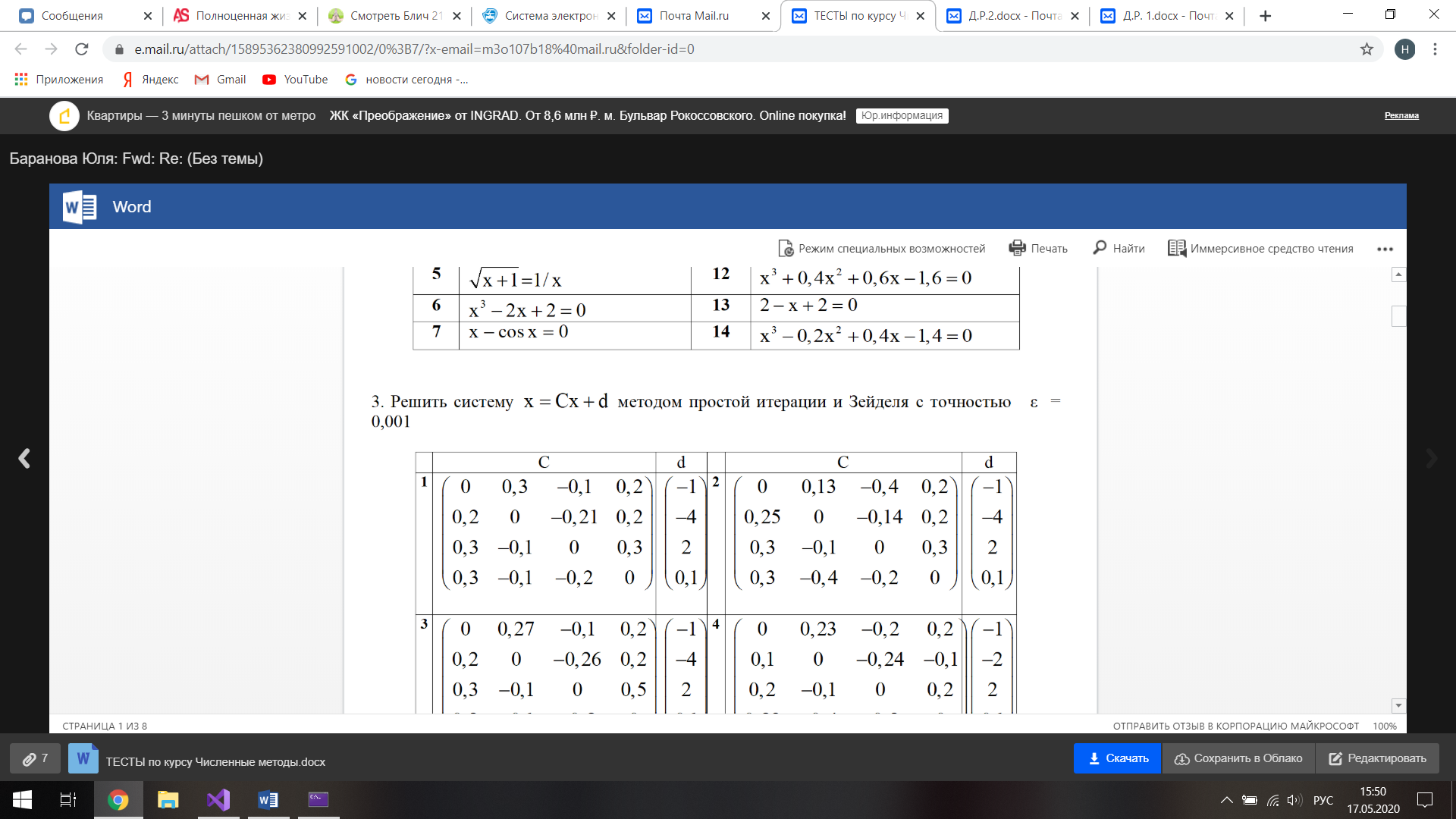
return x;

}

Результат работы программы







Части кода программы

const int m = 4;

double c[m][m] = { 0, 0.13, -0.4, 0.2,

0.25, 0, -0.14, 0.2,

0.3, -0.1, 0, 0.3,

0.3, -0.4, -0.2, 0 };

double d[m] = { -1,-4,2,0.1 };

cout << "\n№ 3.1 " << endl;

double от[m];

ПростойИтерацииСист(c, d, от, 0.001);

cout << "Ответ: x = ( ";

for (int i = 0; i < m; i++)

cout << от[i] << " ";

cout <<" )\n";

cout << "\n\t\t\t№ 3.2 " << endl;

Зойдель(c, d, от, 0.001);

cout << "Ответ: x = ( ";

for (int i = 0; i < m; i++)

cout << от[i] << " ";

cout << " )\n";

void ПростойИтерацииСист(double c[m][m], double d[m], double \*от, double Eps) {

double del= Eps\*2;

double f[m] = { 0 };

bool k = 0;

double x[m];

double x2[m];

for (int i = 0; i < m; i++)

{

x[i] = d[i];

x2[i] = d[i];

}

for (int i = 0; i < m; i++)

for (int j = 0; j < m; j++)

f[i] += abs(c[i][j]);

for (int i = 0; i < m; i++)

if (f[i] > 1)

k = 1;

if (!k)

for (int i = 0; i < m; i++)

cout << f[i]<<"\t";

for (int i = 0; i < m; i++)

f[i] = 0;

if (k)

for (int i = 0; i < m; i++)

for (int j = 0; j < m; j++)

f[i] += abs(c[j][i]);

for (int i = 0; i < m; i++)

if (f[i] < 1)

k = 0;

if (k)

cout << "ошибка\n";

cout << "\n";

for (int i = 0; i < m; i++)

cout << "x[" << i << "]\t";

cout << "del\n";

for (int i = 0; i < m; i++)

printf("%.4f\t", x[i]);

cout << "\n";

while (del > Eps)

{

for (int i = 0; i < m; i++)

{

for (int j = 0; j < m; j++)

x[i] += c[i][j] \* x2[j];

x[i] += d[i];

}

del = 0;

for (int i = 0; i < m; i++)

{

del += abs(x[i] - x2[i]);

printf("%.4f\t", x[i]);

x2[i] = x[i];

x[i] = 0;

}

cout << del << "\n";

}

for (int i = 0; i < m; i++)

от[i] = x2[i];

}

void Зойдель(double c[m][m], double d[m], double \*от, double Eps) {

double del = Eps \* 2;

double f[m] = { 0 };

bool k = 0;

double t=0;

double x[m];

double x2[m];

for (int i = 0; i < m; i++)

{

x[i] = d[i];

x2[i] = d[i];

}

for (int i = 0; i < m; i++)

for (int j = 0; j < m; j++)

f[i] += abs(c[i][j]);

for (int i = 0; i < m; i++)

if (f[i] > 1)

k = 1;

if (!k)

for (int i = 0; i < m; i++)

cout << f[i] << "\t";

for (int i = 0; i < m; i++)

f[i] = 0;

if (k)

for (int i = 0; i < m; i++)

for (int j = 0; j < m; j++)

f[i] += abs(c[j][i]);

for (int i = 0; i < m; i++)

if (f[i] < 1)

k = 0;

if (k)

cout << "ошибка\n";

cout << "\n";

for (int i = 0; i < m; i++)

cout << "x[" << i << "]\t";

cout << "del\n";

for (int i = 0; i < m; i++)

printf("%.4f\t", x[i]);

cout << "\n";

while (del > Eps)

{

for (int i = 0; i < m; i++)

{

for (int j = 0; j < m; j++)

t += c[i][j] \* x[j];

x[i] = t + d[i];

t = 0;

}

del = 0;

for (int i = 0; i < m; i++)

{

del += abs(x[i] - x2[i]);

printf("%.4f\t", x[i]);

x2[i] = x[i];

}

cout << del << "\n";

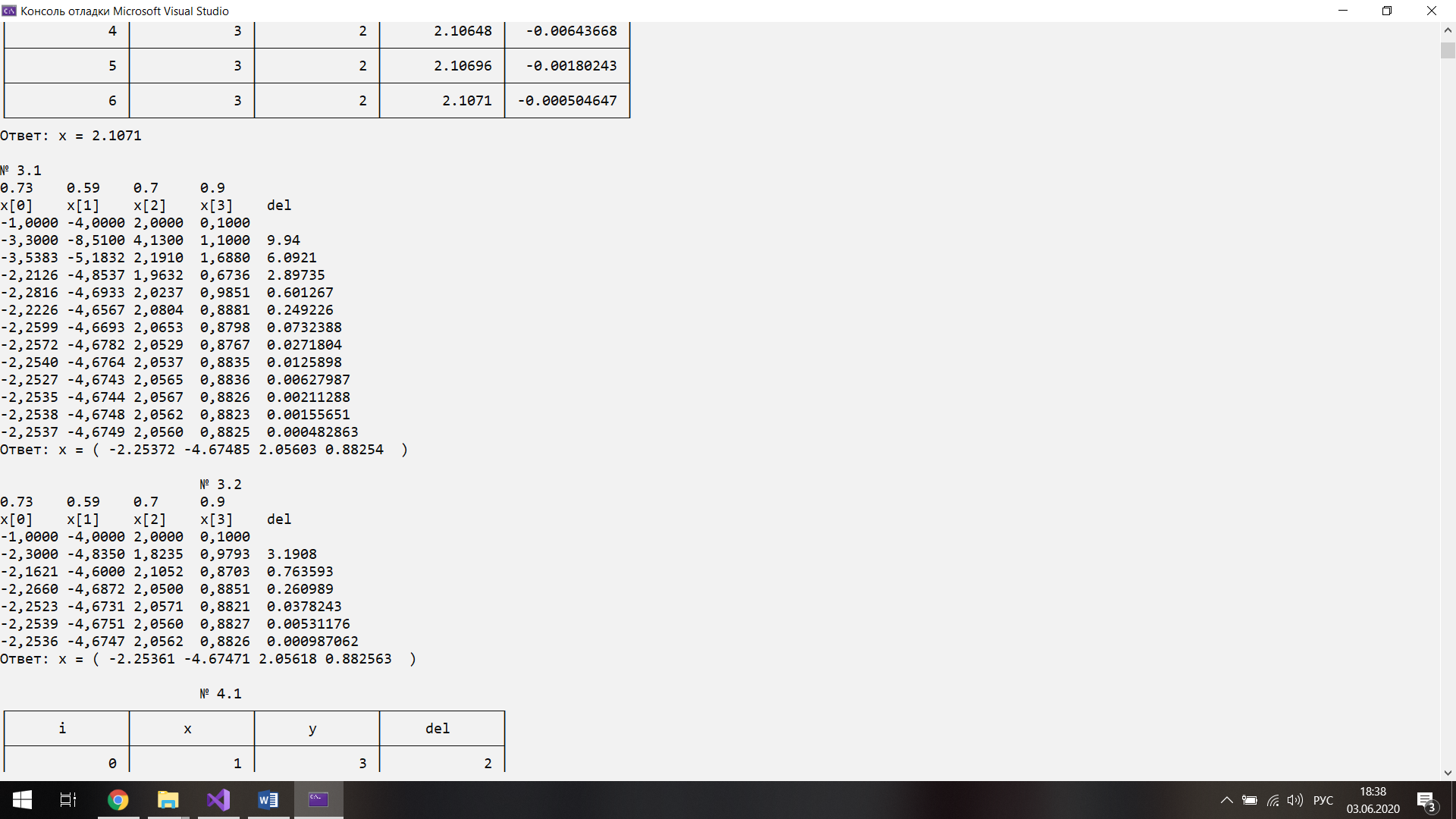
}

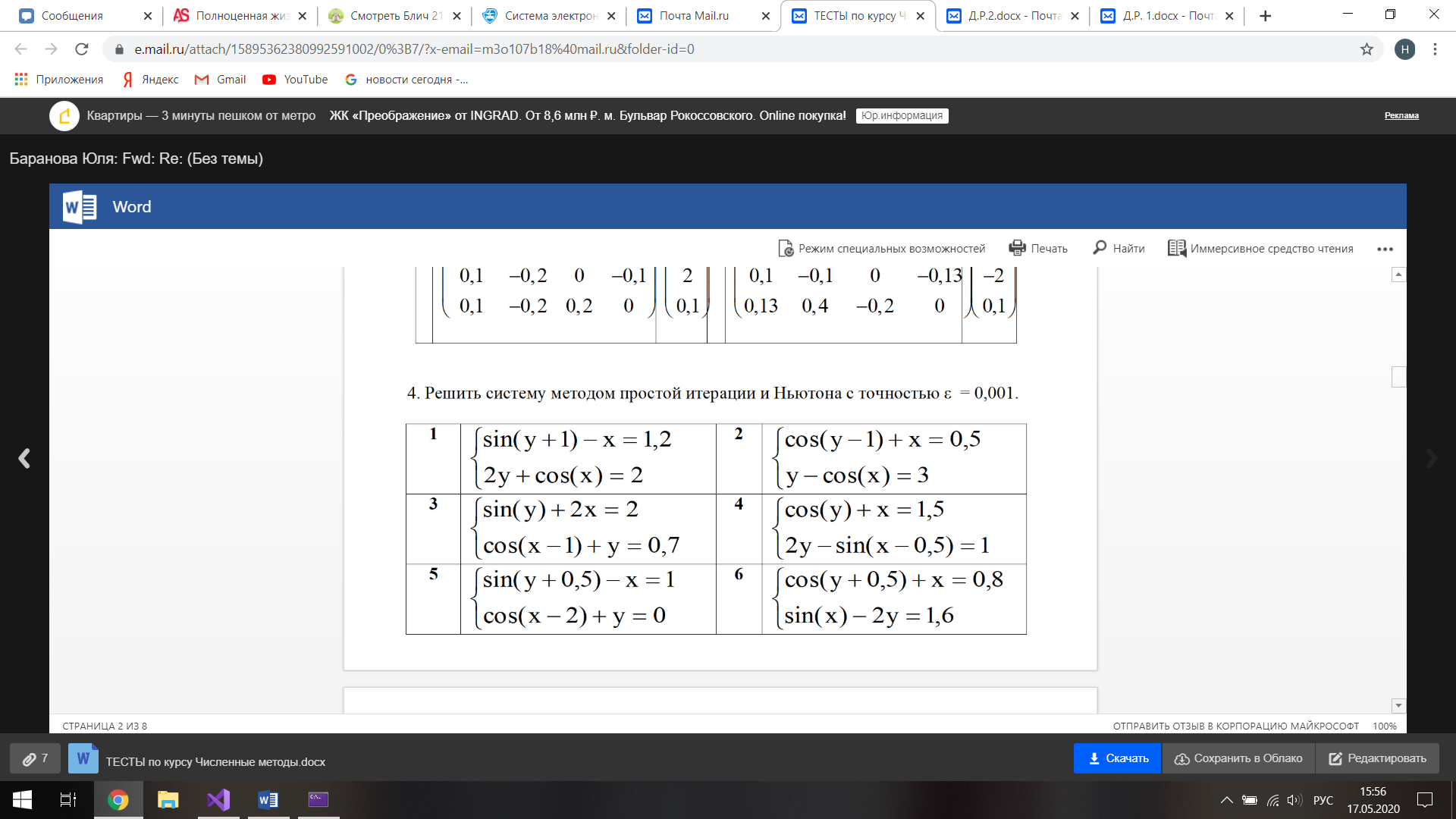
for (int i = 0; i < m; i++)

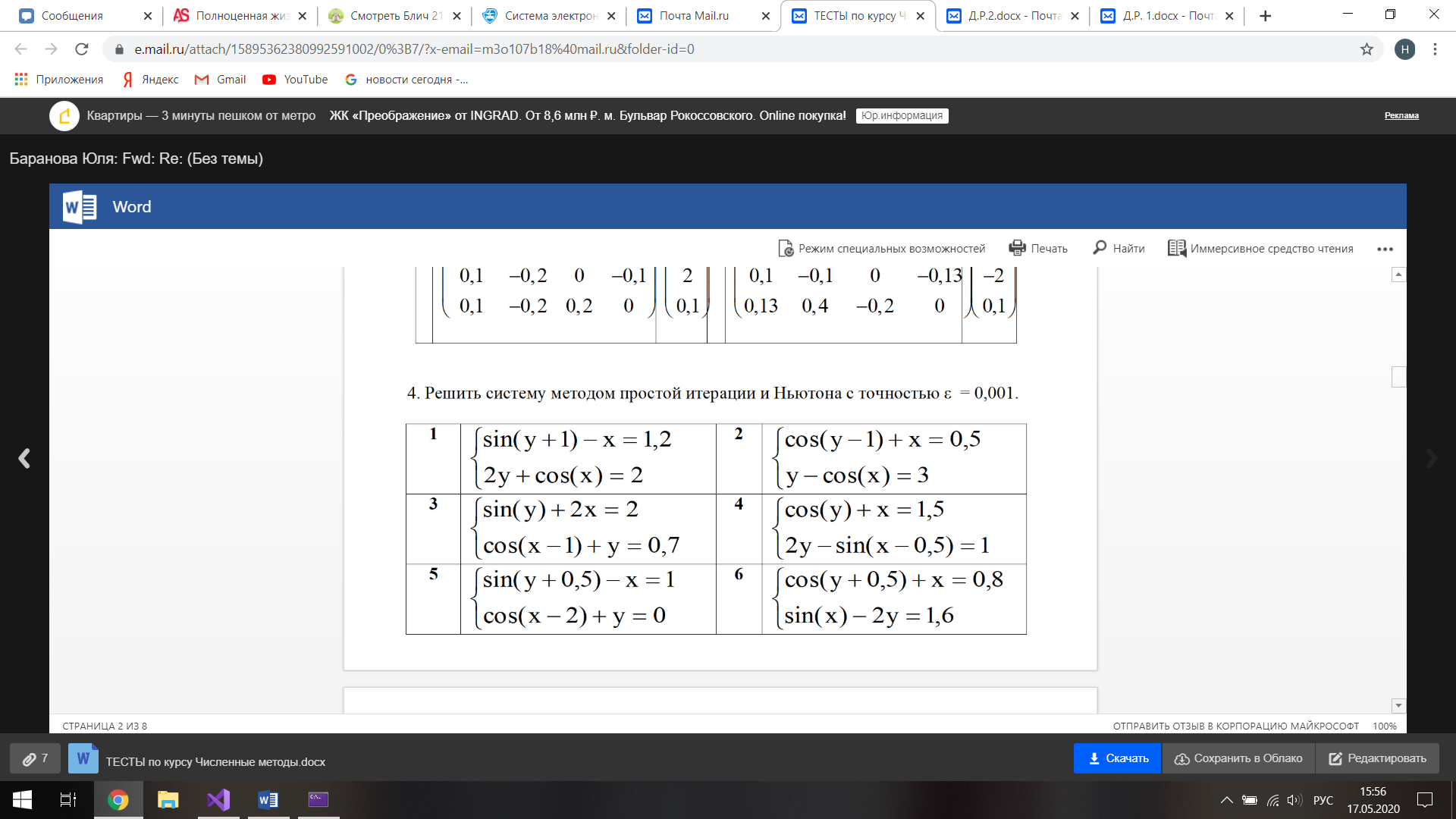
от[i] = x[i];

}

Результат работы программы







Части кода программы

double Fx22(double x, double y) { return 0.5-cos(y-1); }

double Fx32(double x, double y) { return cos(x)+3; }

double Fx23(double x, double y) { return 0.5 - cos(y - 1) - x; }

double Fx33(double x, double y) { return cos(x) + 3 - y; }

double Fx231(double x, double y) { return 1 + sin(x) \* sin(y - 1); }

double Fx232(double x, double y) { return cos(y - 1) + x - 0.5 + sin(y - 1) \* (y - cos(x) - 3); }

double Fx233(double x, double y) { return y - cos(x) - 3 - sin(x) \* (cos(y - 1) + x - 0.5); }

double x = 1;

double y = 3;

cout << "\n\t\t\t№ 4.1 " << endl;

ПростойИтерацииСистНелин(x, y, Fx22, Fx32, 0.001);

cout << "Ответ: (x;y) = (" << x << ";" << y << ")\tf1(x;y) = " << Fx23(x, y) << "\tf2(x;y) = " << Fx33(x, y) << endl;

x = 1; y = 3;

cout << "\n\t\t\t№ 4.2 " << endl;

Ньютон(x, y, Fx231, Fx232, Fx233, 0.001);

cout << "Ответ: (x;y) = (" << x << ";" << y << ")\tf1(x;y) = " << Fx23(x, y) << "\tf2(x;y) = " << Fx33(x, y) << endl;

void ПростойИтерацииСистНелин(double &х, double &у, double (\*F)(double, double), double (\*F2)(double, double), double Eps) {

const int n = 4;

int i = 0;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " x " };

char s2[15] = { " y " };

char s3[15] = { " del " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

N\_tabl(n, s);

N[0] = i;

N[1] = х;

N[2] = у;

double x = х;

double y = у;

double t1 = х-1;

double t2 = у-1;

N[3] = abs(t1 - x) + abs(t2 - y);

while (abs(t1-x) + abs(t2 - y) > Eps)

{

C\_tabl(n, N);

t1 = x;

t2 = y;

i++;

x = F(x, y);

y = F2(x,y);

N[0] = i;

N[1] = x;

N[2] = y;

N[3] = abs(t1 - x) + abs(t2 - y);

}

K\_tabl(n, N);

х = x; у = y;

}

void Ньютон(double& х, double& у, double (\*F1)(double, double), double (\*F2)(double, double), double (\*F3)(double, double), double Eps) {

const int n = 7;

int i = 0;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " x " };

char s2[15] = { " y " };

char s3[15] = { " D " };

char s4[15] = { " Dx " };

char s5[15] = { " Dy " };

char s6[15] = { " del " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

s[4] = s4;

s[5] = s5;

s[6] = s6;

N\_tabl(n, s);

N[0] = i;

N[1] = х;

N[2] = у;

N[3] = F1(х,у);

N[4] = F2(х, у);

N[5] = F3(х, у);

double x = х;

double y = у;

double t1 = х - 1;

double t2 = у - 1;

N[6] = abs(t1 - x) + abs(t2 - y);

while (abs(t1 - x) + abs(t2 - y) > Eps)

{

C\_tabl(n, N);

t1 = x;

t2 = y;

i++;

x = t1 - F2(x, y)/ F1(x, y);

y = t2 - F3(t1, y) / F1(t1, y);

N[0] = i;

N[1] = x;

N[2] = y;

N[3] = F1(x, y);

N[4] = F2(x, y);

N[5] = F3(x, y);

N[6] = abs(t1 - x) + abs(t2 - y);

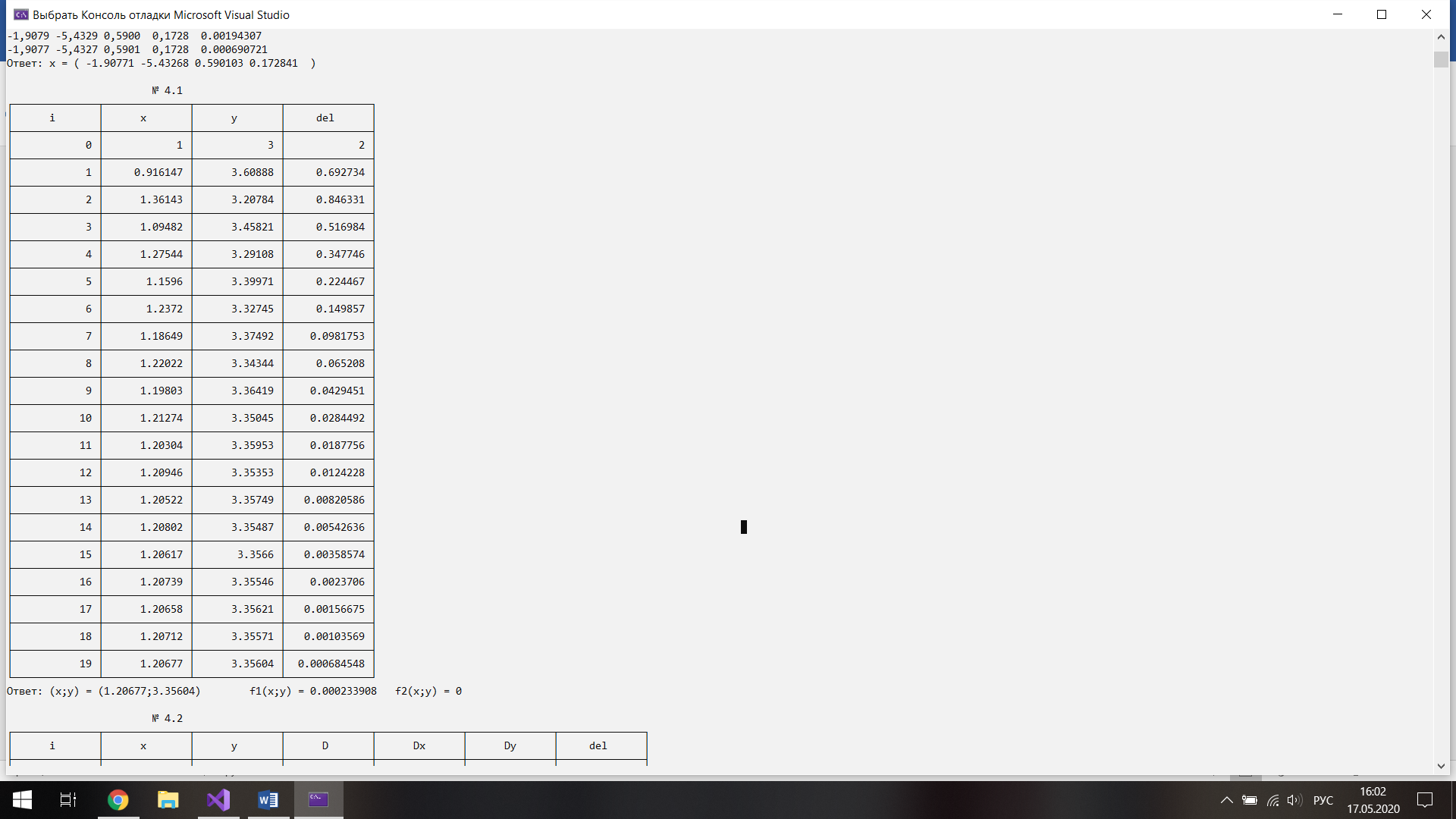
}

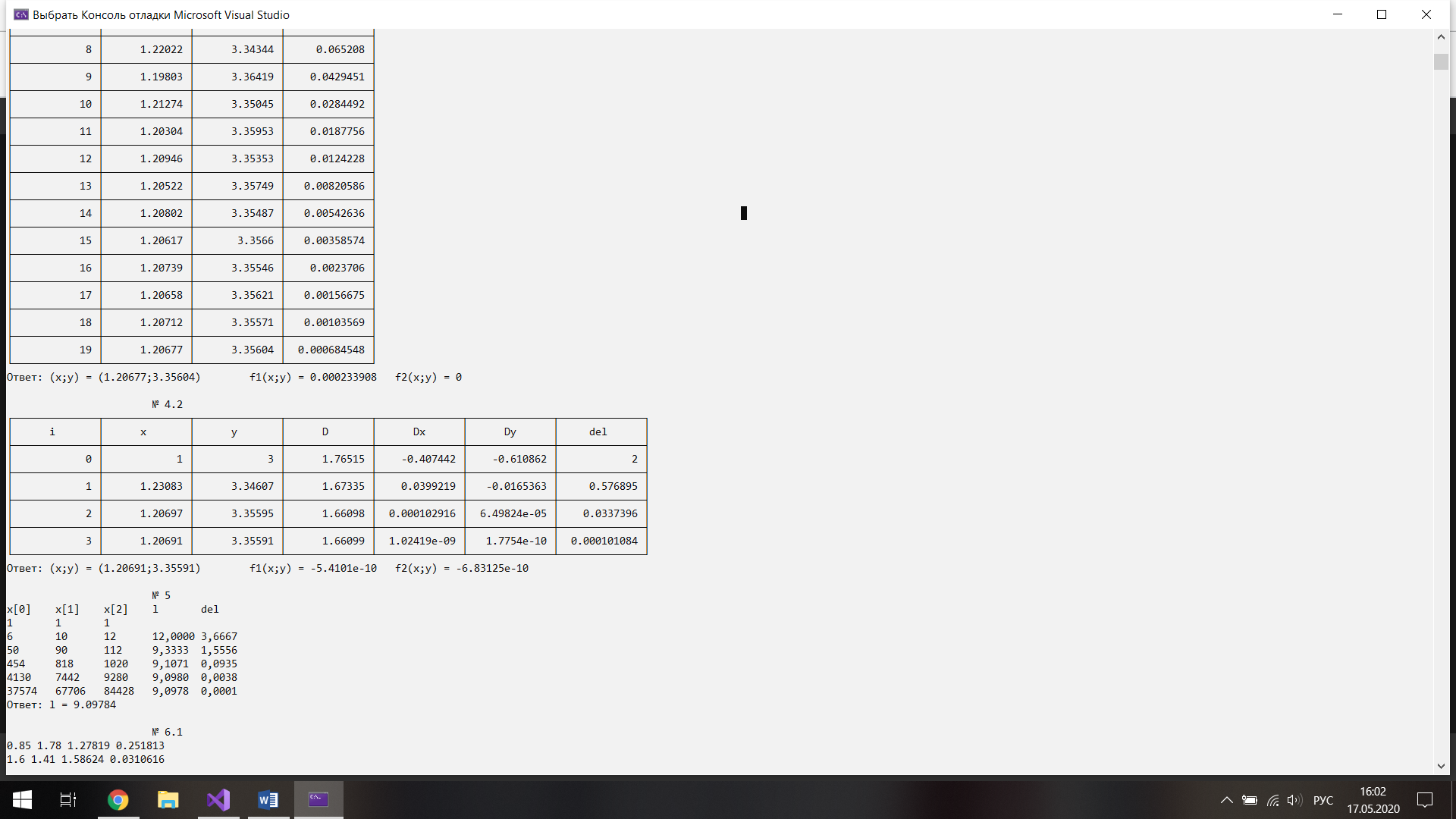
K\_tabl(n, N);

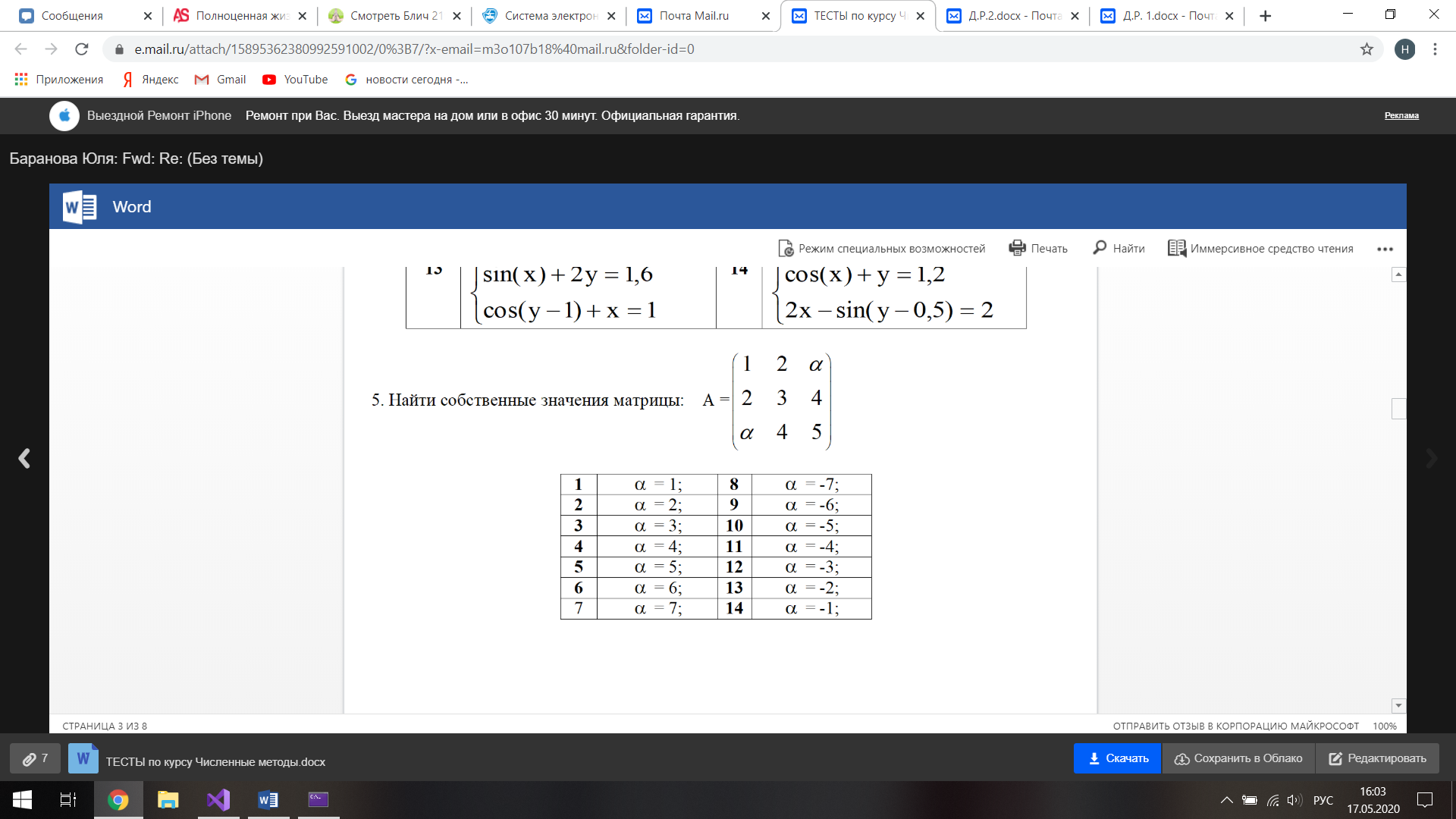
х = x; у = y;

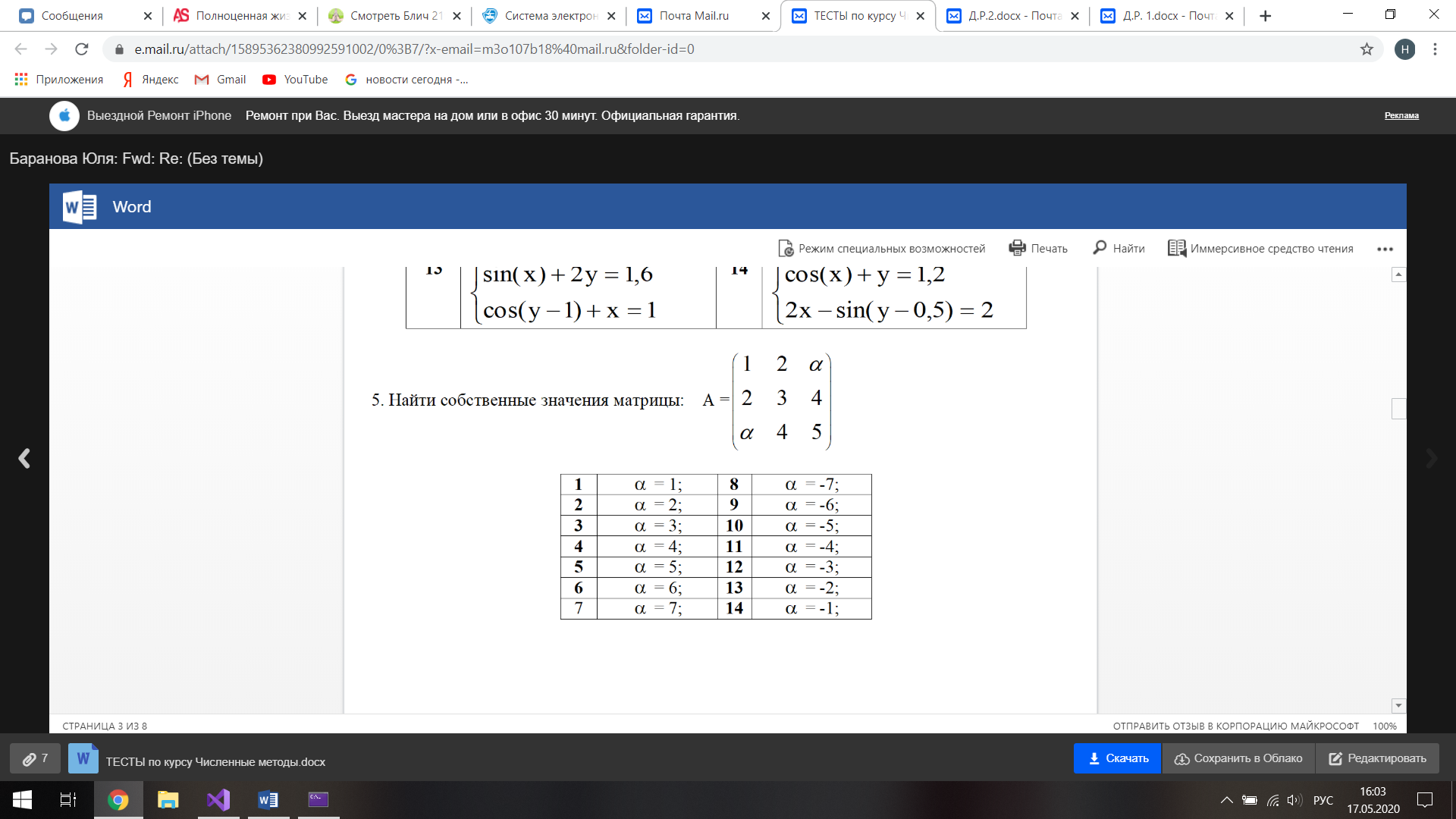
}

Результат работы программы









Части кода программы

const int m1 = 3;

double c1[m1][m1] = { 1, 2, 2,

2, 3, 4,

2, 4, 5, };

double d1[m1] = { 1,1,1 };

cout << "\n\t\t\t№ 5 (Метод вращения) " << endl;

СобствЧислаВращение(c1, d1, 0.001);

cout << "Ответ: (a1;a2;a3) = (" << d1[0] << ";" << d1[1] << ";" << d1[2] << ")" << endl;

void СобствЧислаВращение(double c[m1][m1], double\* d, double Eps) {

double c2[m1][m1];

double H[m1][m1];

double t[m1][m1];

double t2[m1][m1];

double fi;

double max = 1.1\*Eps;

int и = 0;

int жи = 1;

int in = 0;

for (int i = 0; i < m1; i++)

for (int j = 0; j < m1; j++)

c2[i][j] = c[i][j];

for (int i = 0; i < m1 - 1; i++)

for (int j = 0; j < m1; j++)

{

while (i >= j) j++;

if (c2[i][j] > max)

{

max = c2[i][j];

и = i;

жи = j;

}

}

while (max > Eps)

{

in++;

for (int i = 0; i < m1; i++)

for (int j = 0; j < m1; j++)

{

H[i][j] = 0;

t[i][j] = 0;

t2[i][j] = 0;

}

for (int i = 0; i < m1; i++) H[i][i] = 1;

fi = 0.5 \* atan(2 \* c2[и][жи] / (c2[и][и] - c2[жи][жи]));

cout << "\nfi = " << fi << endl;

H[и][и] = cos(fi);

H[жи][жи] = cos(fi);

H[и][жи] = -sin(fi);

H[жи][и] = sin(fi);

cout << "max = " << max << endl;

cout <<"H["<<in<<"]"<< endl;

for (int i = 0; i < m1; i++)

{

for (int j = 0; j < m1; j++)

cout << H[i][j] <<" ";

cout << endl;

}

for (int i = 0; i < m1; i++)

for (int j = 0; j < m1; j++)

for (int k = 0; k < m1; k++)

t[i][j] += H[k][i] \* c2[k][j];

for (int i = 0; i < m1; i++)

for (int j = 0; j < m1; j++)

for (int k = 0; k < m1; k++)

t2[i][j] += t[i][k] \* H[k][j];

for (int i = 0; i < m1; i++)

for (int j = 0; j < m1; j++)

c2[i][j] = t2[i][j];

cout << "A[" << in << "]" << endl;

for (int i = 0; i < m1; i++)

{

for (int j = 0; j < m1; j++)

cout << c2[i][j] << " ";

cout << endl;

}

max = -100;

for (int i = 0; i < m1 - 1; i++)

for (int j = 0; j < m1; j++)

{

while (i >= j) j++;

if (c2[i][j] > max)

{

max = c2[i][j];

и = i;

жи = j;

}

}

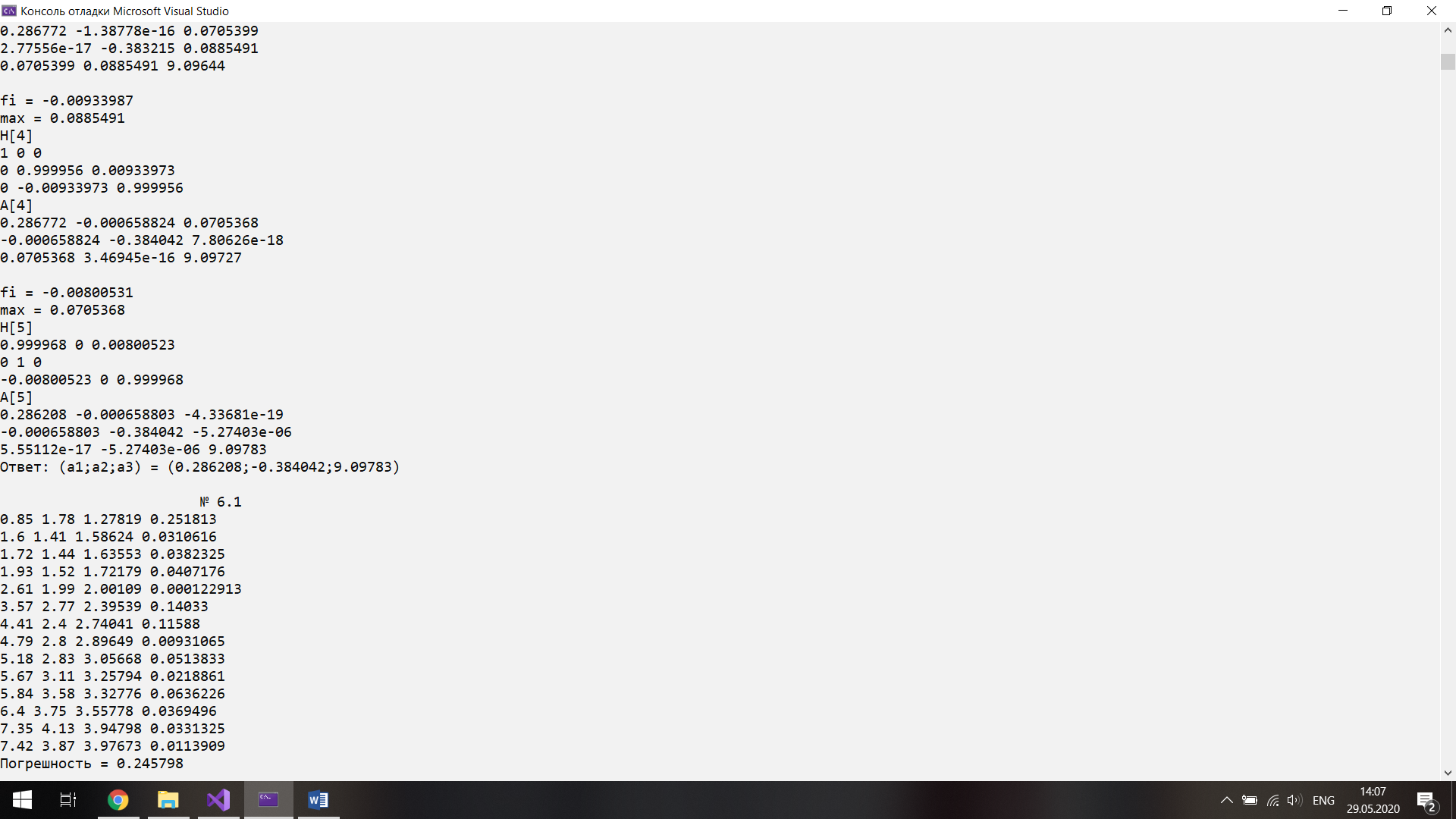
}

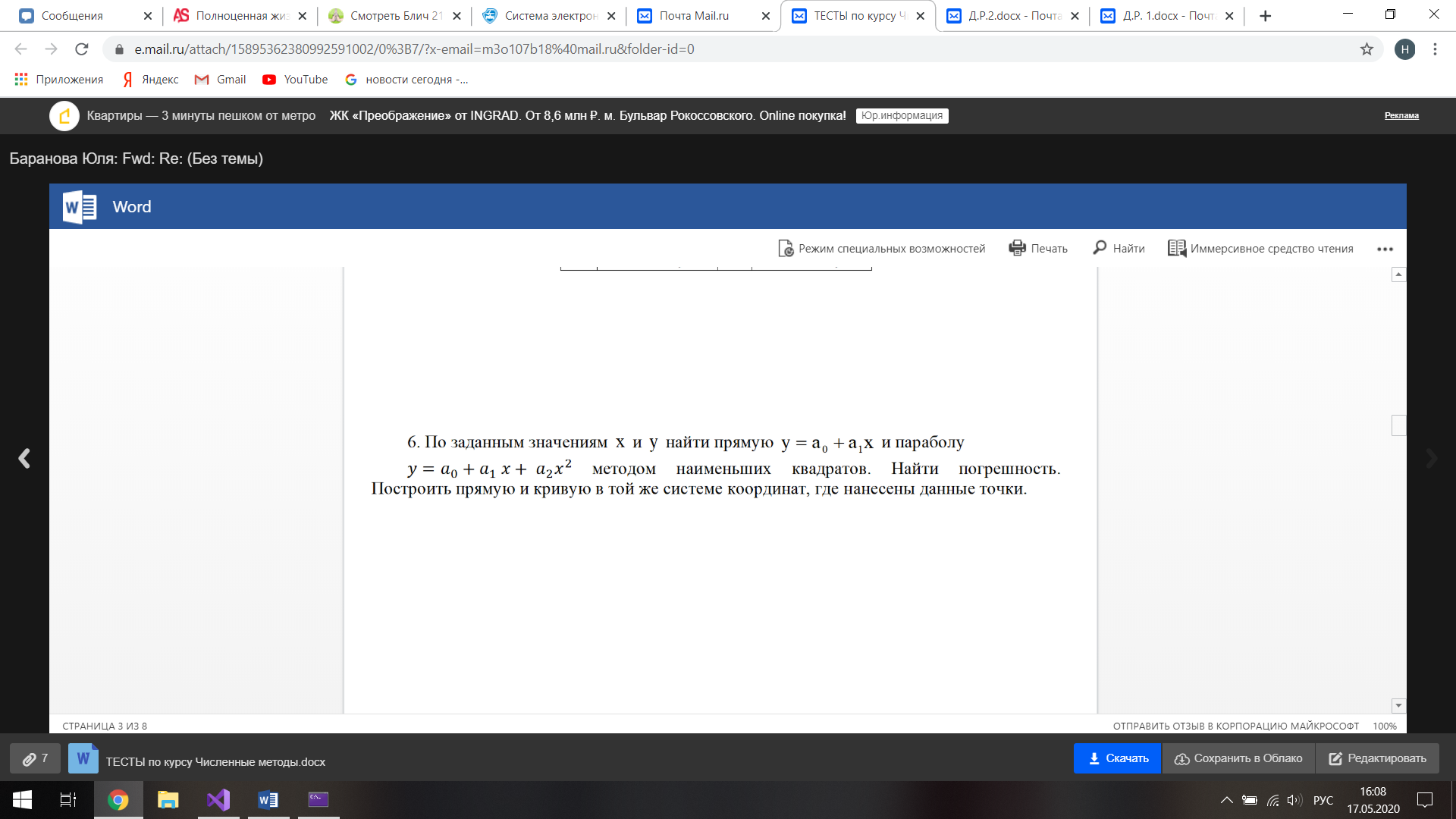
for (int i = 0; i < m1; i++)

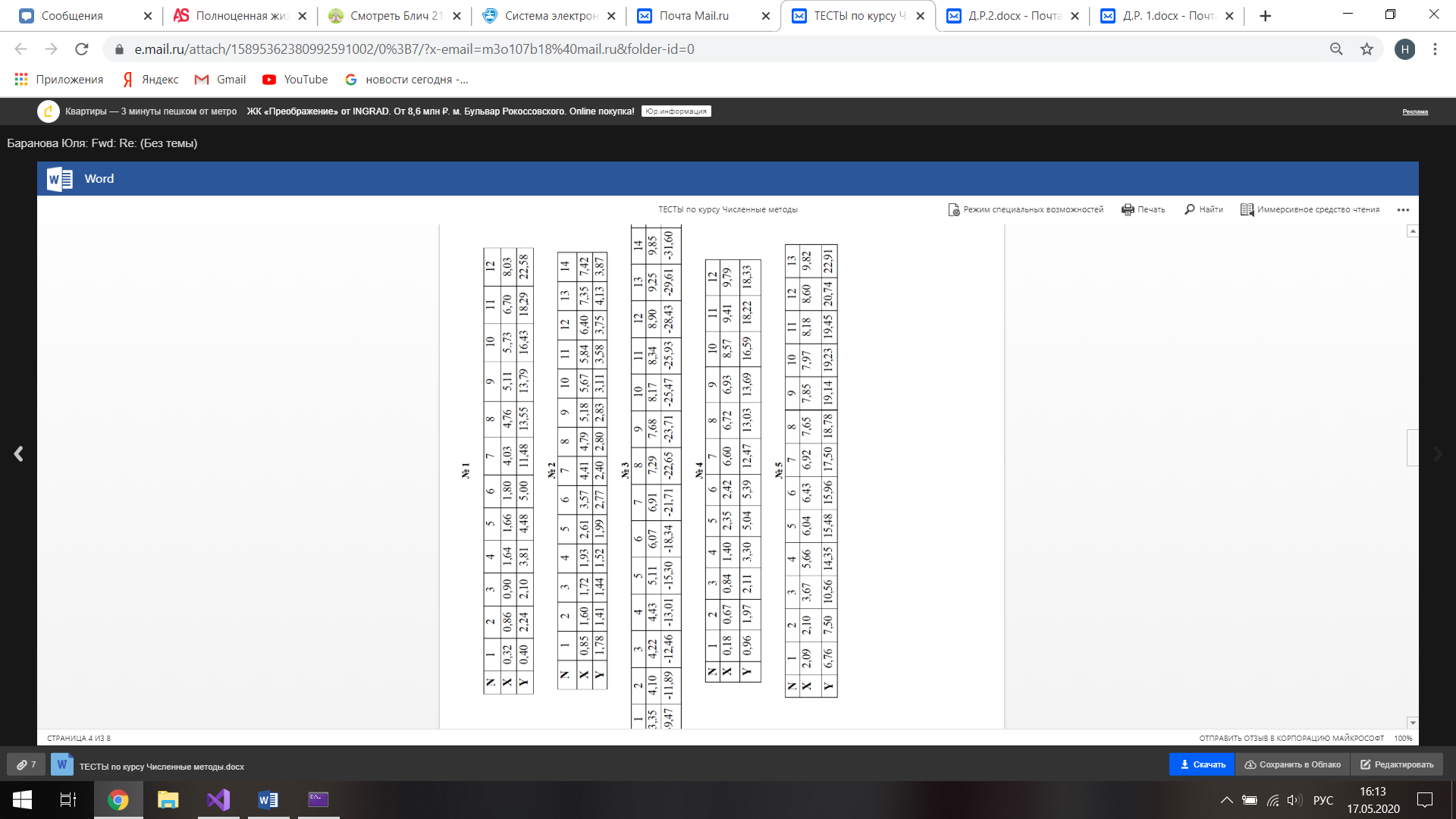
d[i] = c2[i][i];

}

Результат работы программы







Части кода программы

double х[14] = { 0.85,1.6, 1.72,1.93,2.61,3.57,4.41,4.79,5.18,5.67,5.84,6.4, 7.35,7.42 };

double у[14] = { 1.78,1.41,1.44,1.52,1.99,2.77,2.4, 2.8, 2.83,3.11,3.58,3.75,4.13,3.87 };

double ab[3];

cout << "\n\t\t\t№ 6.1 " << endl;

НаимКвЛин(х, у, ab, 14);

cout << "Ответ: (а;b) = (" << ab[1] << ";" << ab[0] << ")" << endl;

cout << "\n\t\t\t№ 6.2 " << endl;

НаимКвЛин(х, у, ab, 14,0);

cout << "Ответ: (а;b;c) = (" << ab[2] << ";" << ab[1] << ";" << ab[0] << ")" << endl;

void НаимКвЛин(double\* х, double\* у, double\* a, int k, bool t) {

double t1 = 0;

double x = 0;

double x2 = 0;

double x3 = 0;

double x4 = 0;

double y = 0;

double xy = 0;

double x2y = 0;

for (size\_t i = 0; i < k; i++)

{

x += х[i];

x2 += х[i] \* х[i];

x3 += х[i] \* х[i] \* х[i];

x4 += х[i] \* х[i] \* х[i] \* х[i];

y += у[i];

xy += у[i] \* х[i];

x2y += у[i] \* х[i] \* х[i];

}

if (t) {

a[1] = (k \* xy - x \* y) / (k \* x2 - x \* x);

a[0] = (y - a[1] \* x) / k;

for (size\_t i = 0; i < k; i++)

{

t1 += pow(х[i] \* a[1] + a[0] - у[i],2);

cout << х[i] << " " << у[i] << " " << х[i] \* a[1] + a[0] << " " << pow(х[i] \* a[1] + a[0] - у[i], 2) << "\n";

}

}

else {

a[0] = (y \* x2 \* x4 + x \* x3 \* x2y + xy \* x3 \* x2 - x2 \* x2 \* x2y - x \* xy \* x4 - x3 \* x3 \* y) / (k \* x2 \* x4 + x \* x3 \* x2 + x \* x3 \* x2 - x2 \* x2 \* x2 - x \* x \* x4 - x3 \* x3 \* k);

a[1] = (k \* xy \* x4 + y \* x3 \* x2 + x \* x2y \* x2 - x2 \* xy \* x2 - y \* x \* x4 - x3 \* x2y \* k) / (k \* x2 \* x4 + x \* x3 \* x2 + x \* x3 \* x2 - x2 \* x2 \* x2 - x \* x \* x4 - x3 \* x3 \* k);

a[2] = (k \* x2 \* x2y + x \* xy \* x2 + x \* x3 \* y - y \* x2 \* x2 - x \* x \* x2y - xy \* x3 \* k) / (k \* x2 \* x4 + x \* x3 \* x2 + x \* x3 \* x2 - x2 \* x2 \* x2 - x \* x \* x4 - x3 \* x3 \* k);

for (size\_t i = 0; i < k; i++)

{

t1 += pow(х[i] \* х[i] \* a[2] + х[i] \* a[1] + a[0] - у[i],2);

cout << х[i] << " " << у[i] << " " << х[i] \* х[i] \* a[2] + х[i] \* a[1] + a[0] << " " << pow(х[i] \* х[i] \* a[2] + х[i] \* a[1] + a[0] - у[i], 2) << "\n";

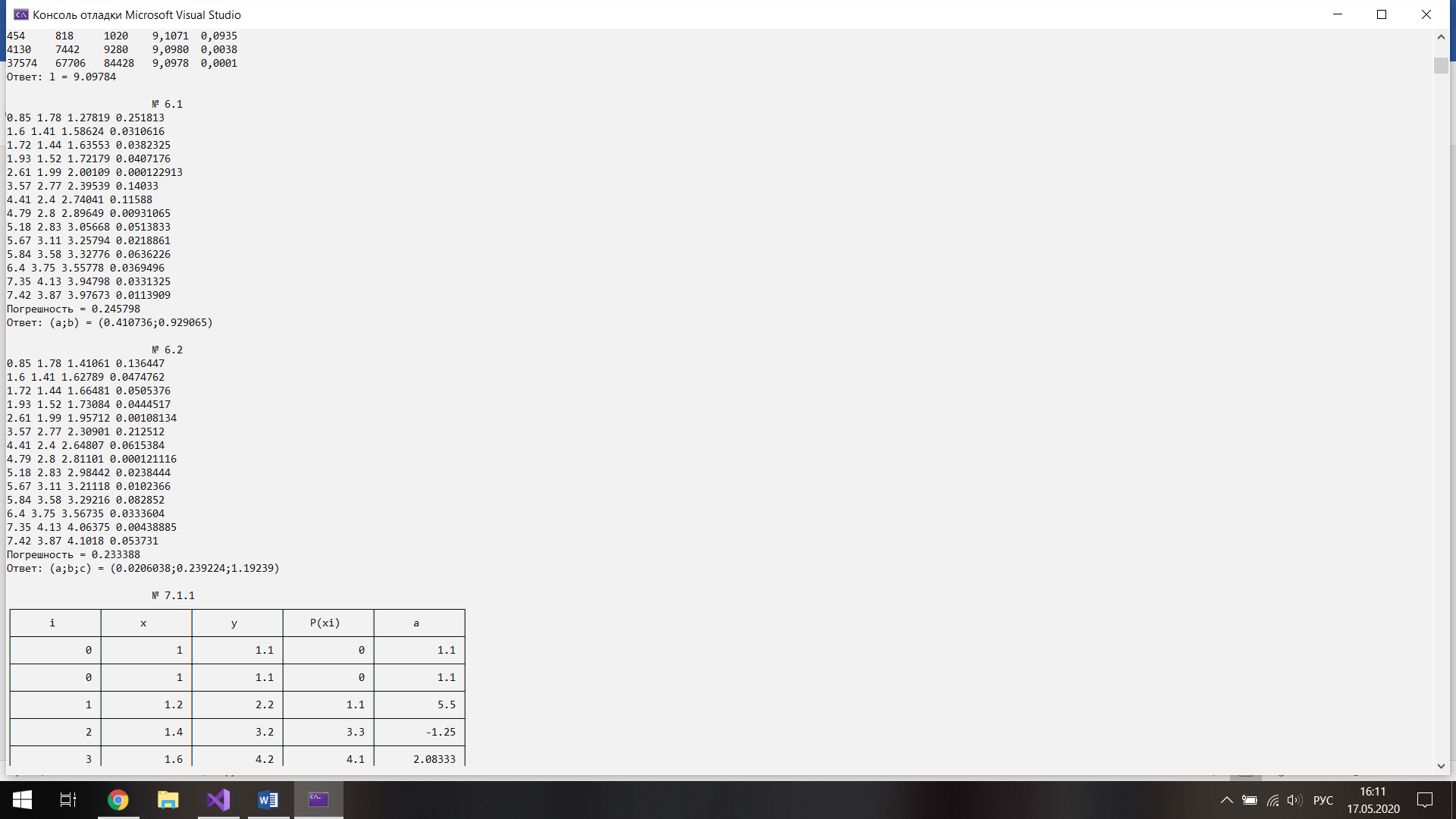
}

}

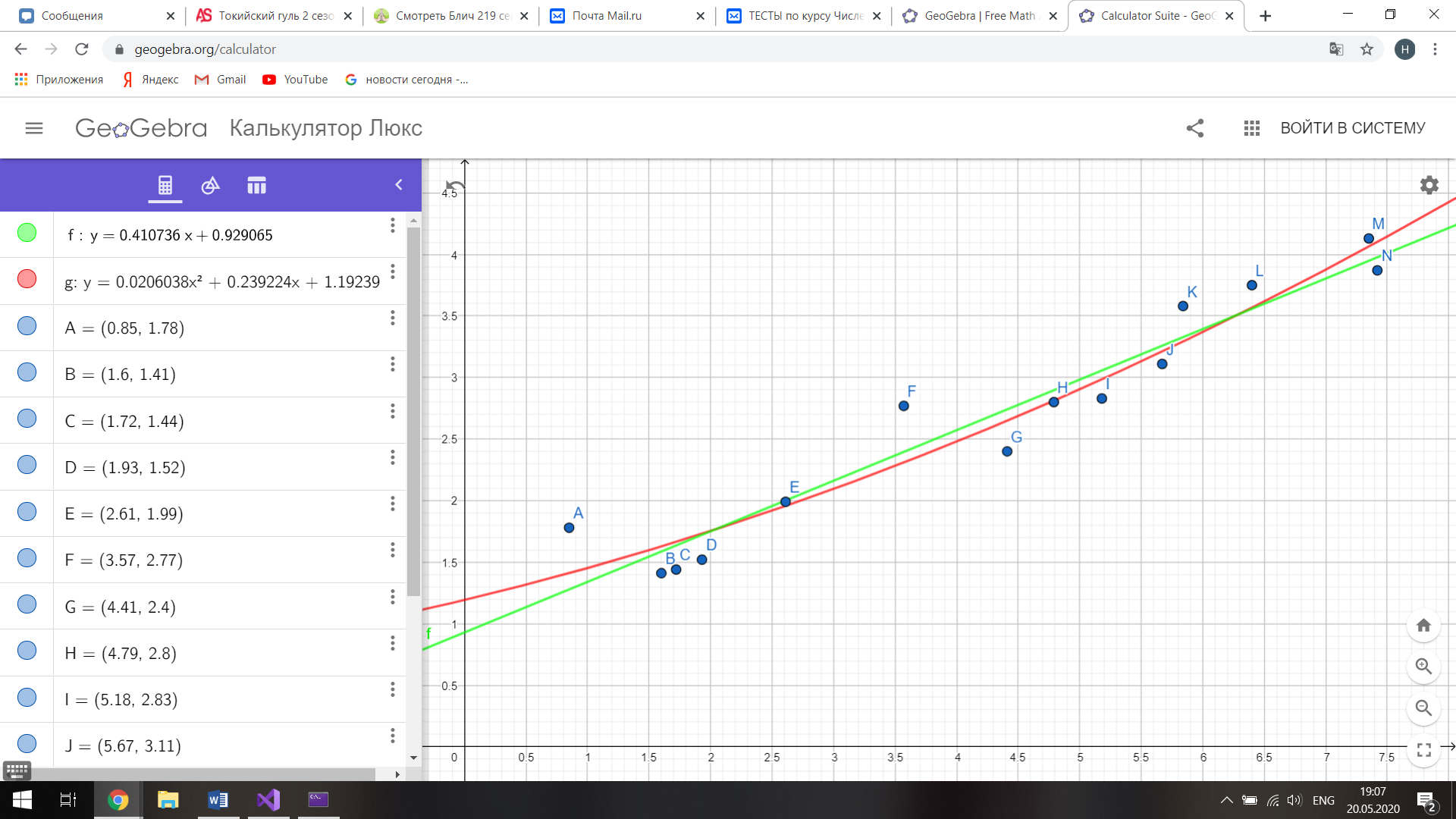
cout << "Погрешность = "<< pow((t1/k),0.5) <<"\n";

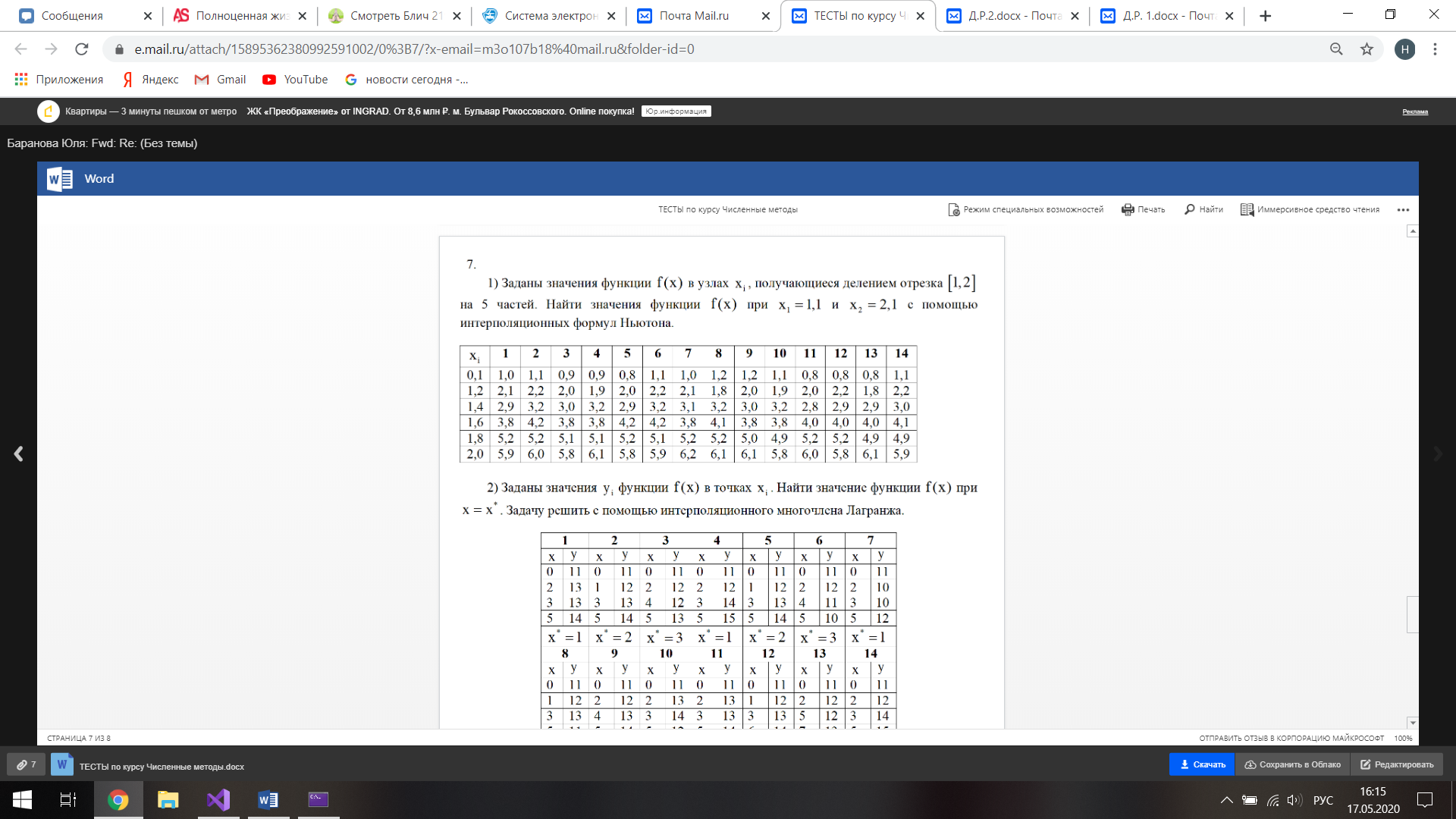
}

Результат работы программы



Прямая и кривая, построенные в той же системе координат, где нанесены данные точки.





Части кода программы

const int m2 = 4;

double х1[m2] = { 0, 1, 3, 5 };

double у1[m2] = { 11,12,13,14 };

const int m3 = 6;

double х2[m3] = { 1, 1.2, 1.4, 1.6, 1.8, 2 };

double у2[m3] = { 1.1, 2.2, 3.2, 4.2, 5.2, 6 };

double ab2[m3];

cout << "\n\t\t\t№ 7.1.1 " << endl;

cout << "Ответ: y(1.1) = " << ИнтерполяцияНьютон(х2, у2, ab2, m3, 1.1) << endl;

cout << "\n\t\t\t№ 7.1.2 " << endl;

cout << "Ответ: y(2) = " << ИнтерполяцияНьютон(х2, у2, ab2, m3, 2) << endl;

double ab3[m2];

cout << "\n\t\t\t№ 7.2 " << endl;

cout << "Ответ: y(2.1) = " << ИнтерполяцияЛогранж(х1, у1, ab3, m2, 2.1) << endl;

double Px(double\* x, double\* a, int k, double t)

{

double S = 0;

double P = 1;

for (size\_t i = 0; i < k; i++)

{

for (size\_t j = 0; j < i; j++)

P = P \* (t - x[j]);

S = S + a[i]\*P;

P = 1;

}

return S;

}

double ИнтерполяцияНьютон(double\* x, double\* y, double\* a, int k, double t) {

const int n = 5;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " x " };

char s2[15] = { " y " };

char s3[15] = { " P(xi) " };

char s4[15] = { " a " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

s[4] = s4;

N\_tabl(n, s);

N[0] = 0;

N[1] = x[0];

N[2] = y[0];

N[3] = Px(x, a, 0, x[0]);

N[4] = y[0];

double S = 0;

double P = 1;

for (size\_t i = 0; i < k; i++)

a[i] = 0;

for (size\_t i = 0; i < k; i++)

{

C\_tabl(n, N);

for (size\_t j = 0; j < i; j++)

P = P \* (x[i] - x[j]);

a[i] = (y[i]- Px(x, a, i, x[i])) / P;

P = 1;

N[0] = i;

N[1] = x[i];

N[2] = y[i];

N[3] = Px(x, a, i, x[i]);

N[4] = a[i];

}

K\_tabl(n, N);

return Px(x, a, k, t);

}

double ИнтерполяцияЛогранж(double\* x, double\* y, double\* a, int k, double t) {

double S = 0;

double P = 1;

for (size\_t i = 0; i < k; i++)

{

for (size\_t j = 0; j < k; j++)

if (i != j)

P \*= (t - x[j]) / (x[i] - x[j]);

cout << "L" << i << "(" << t << ") = " << P << " ";

S += y[i] \* P;

P = 1;

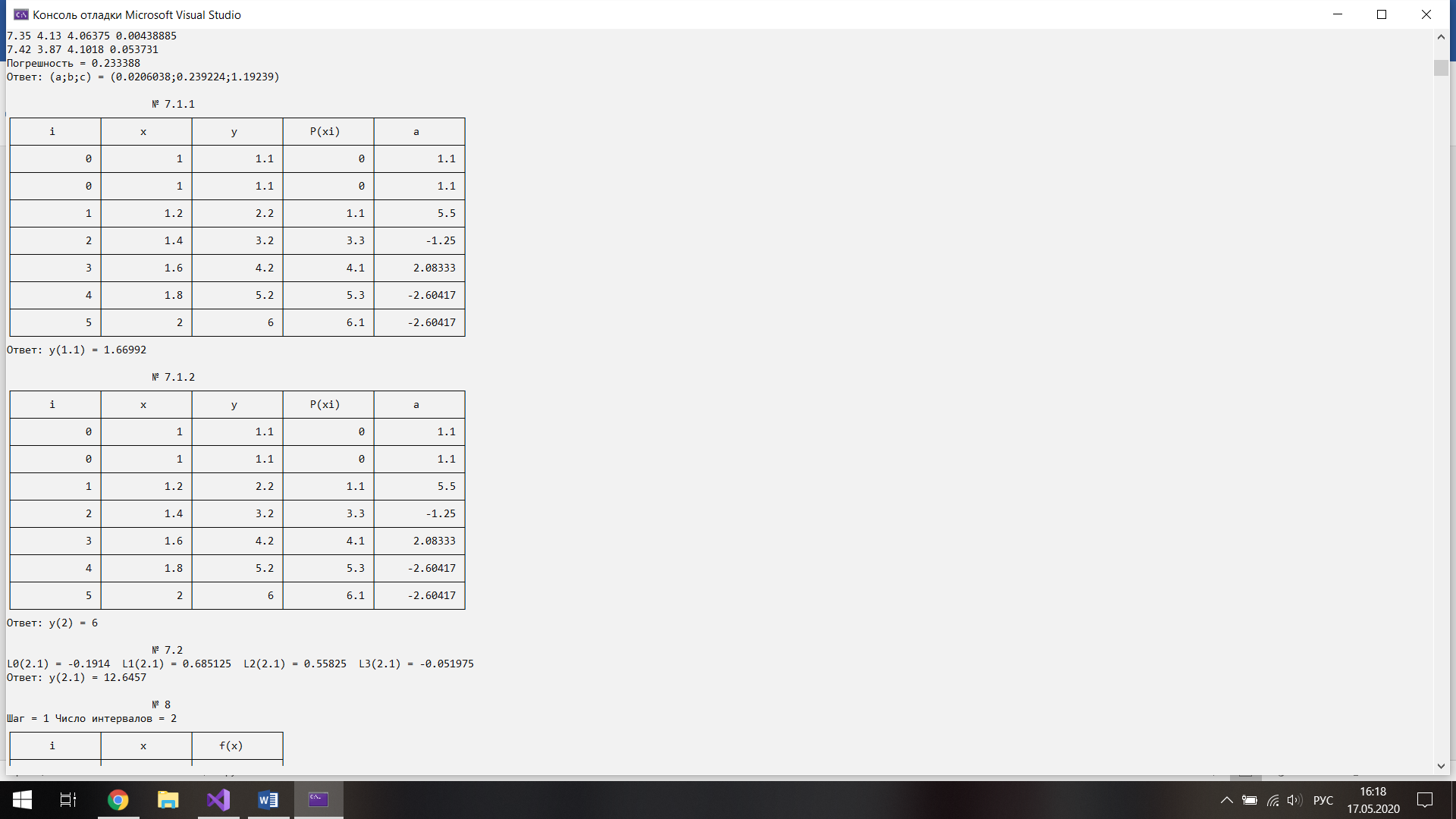
}

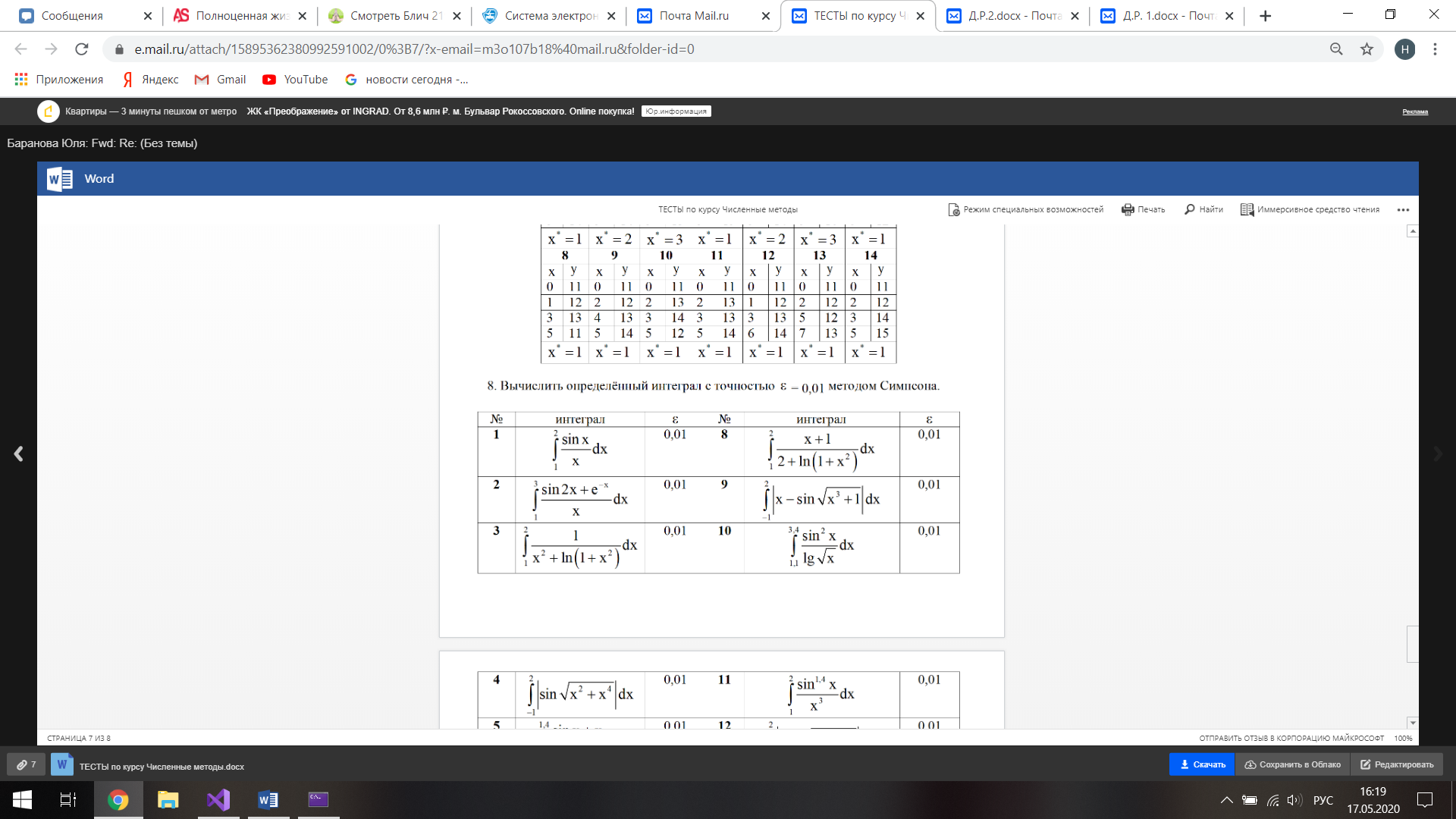
cout << "\n";

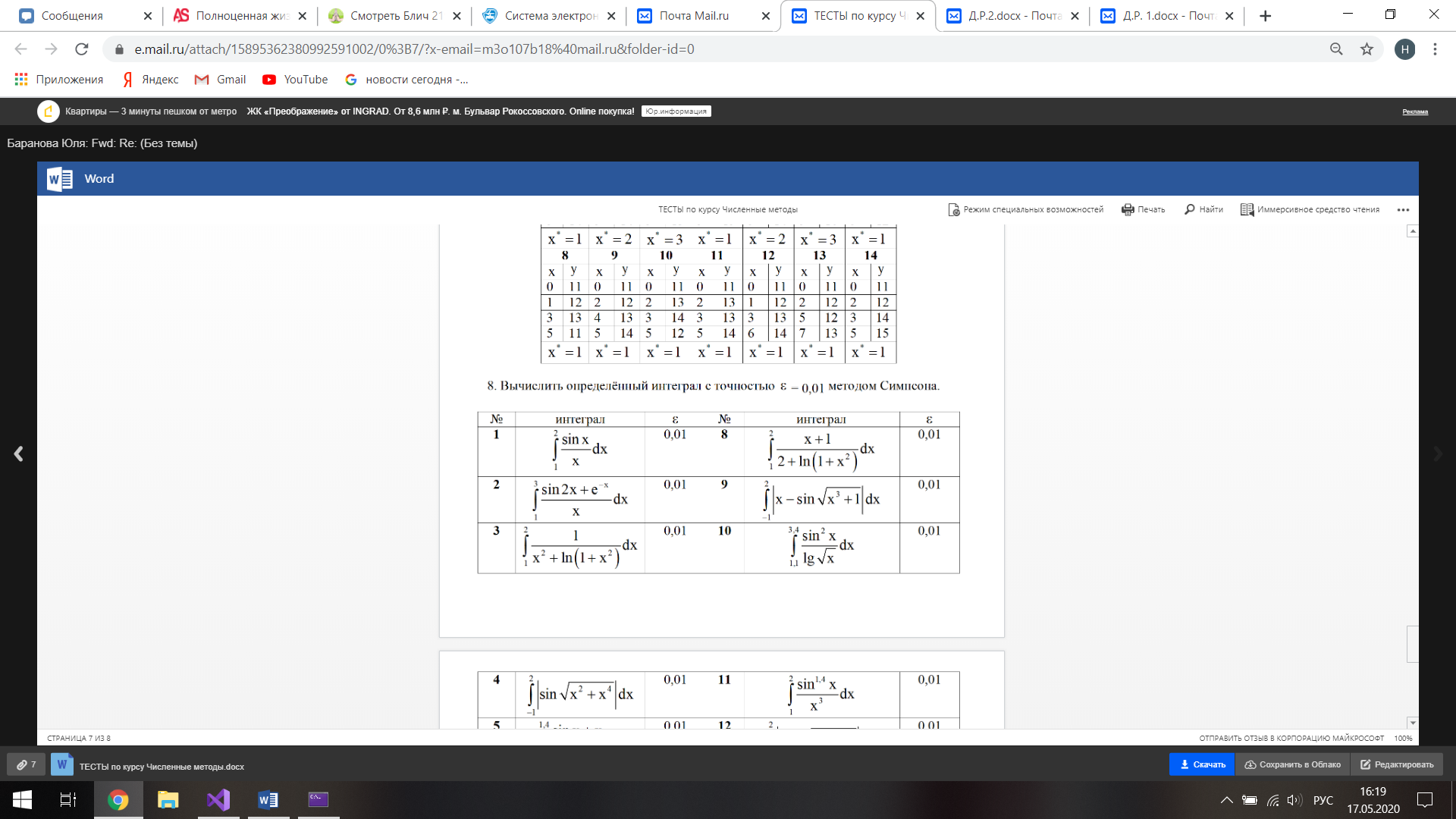
return S;

}

Результат работы программы







Части кода программы

double Fx24(double x) { return (sin(2 \* x) + pow(2.7182818284, -x)) / x; }

cout << "\n\t\t\t№ 8 " << endl;

cout << "Ответ: " << ИнтегралСимпсон(1, 3, Fx24, 0.01) << endl;

double ИнтегралСимпсон(double a, double b, double (\*F)(double), double Eps) {

const int n = 3;

int i = 0;

double N[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " x " };

char s2[15] = { " f(x) " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

N[0] = i;

N[1] = a;

N[2] = F(a);

double del;

double l=100;

double l2=0;

while (abs(l-l2)/15 > Eps)

{

l2 = l;

l = 0;

i++;

del = (b - a) / double(2 \* i);

cout << "Шаг = " << del << " Число интервалов = " << i \* 2 << endl;

N\_tabl(n, s);

for (int j = 0; j <= 2\*i; j++)

{

if(j>0)

C\_tabl(n, N);

if (j == 0 || j == 2 \* i)

l += F(a + j \* del);

else if (j % 2 == 1)

l += 4 \* F(a + j \* del);

else

l += 2 \* F(a + j \* del);

N[0] = j;

N[1] = a + j \* del;

N[2] = F(a + j \* del);

}

K\_tabl(n, N);

l = del / 3 \* l;

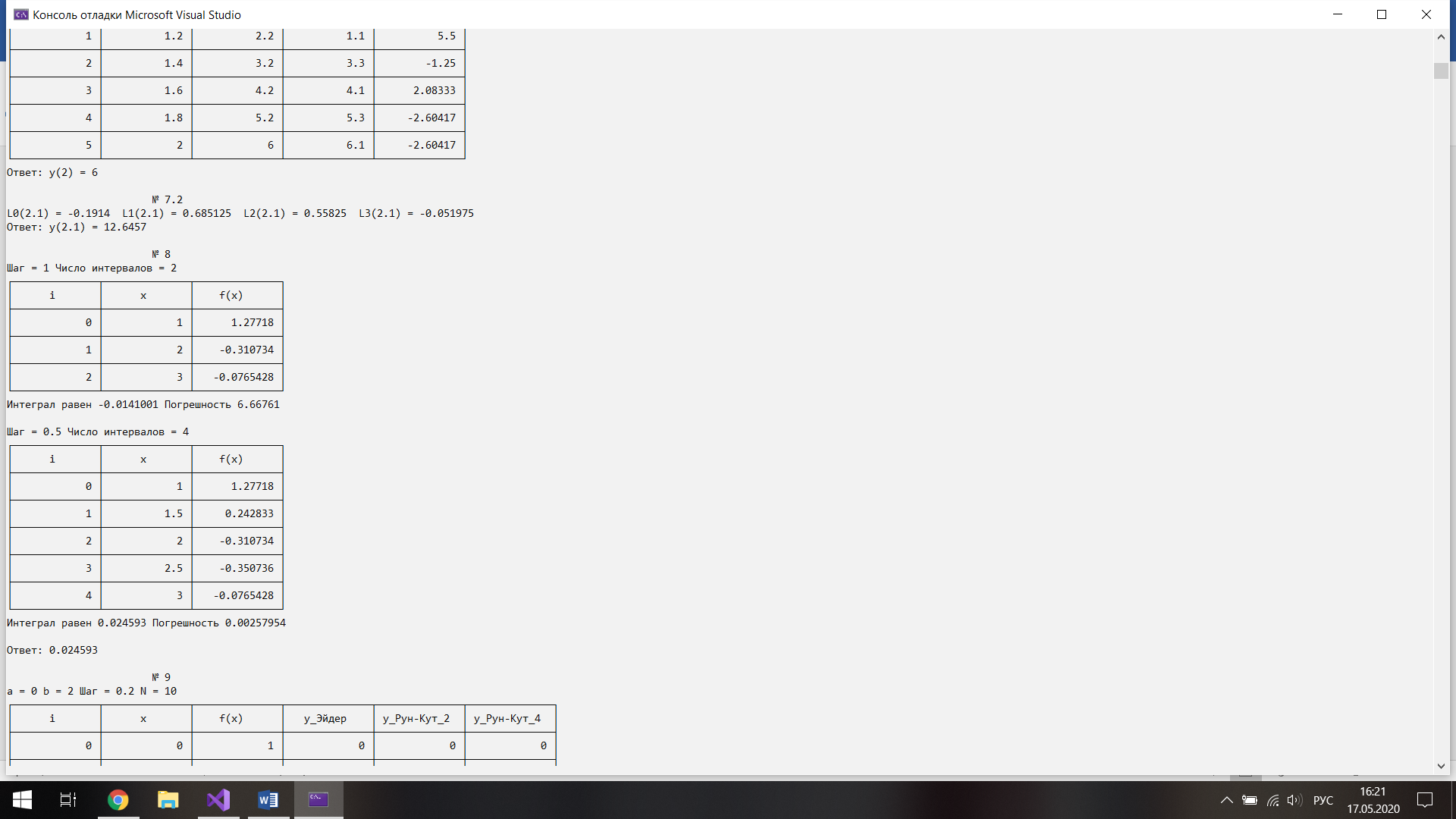
cout << "Интеграл равен " << l << " Погрешность " << abs(l - l2) / 15 << endl << endl;

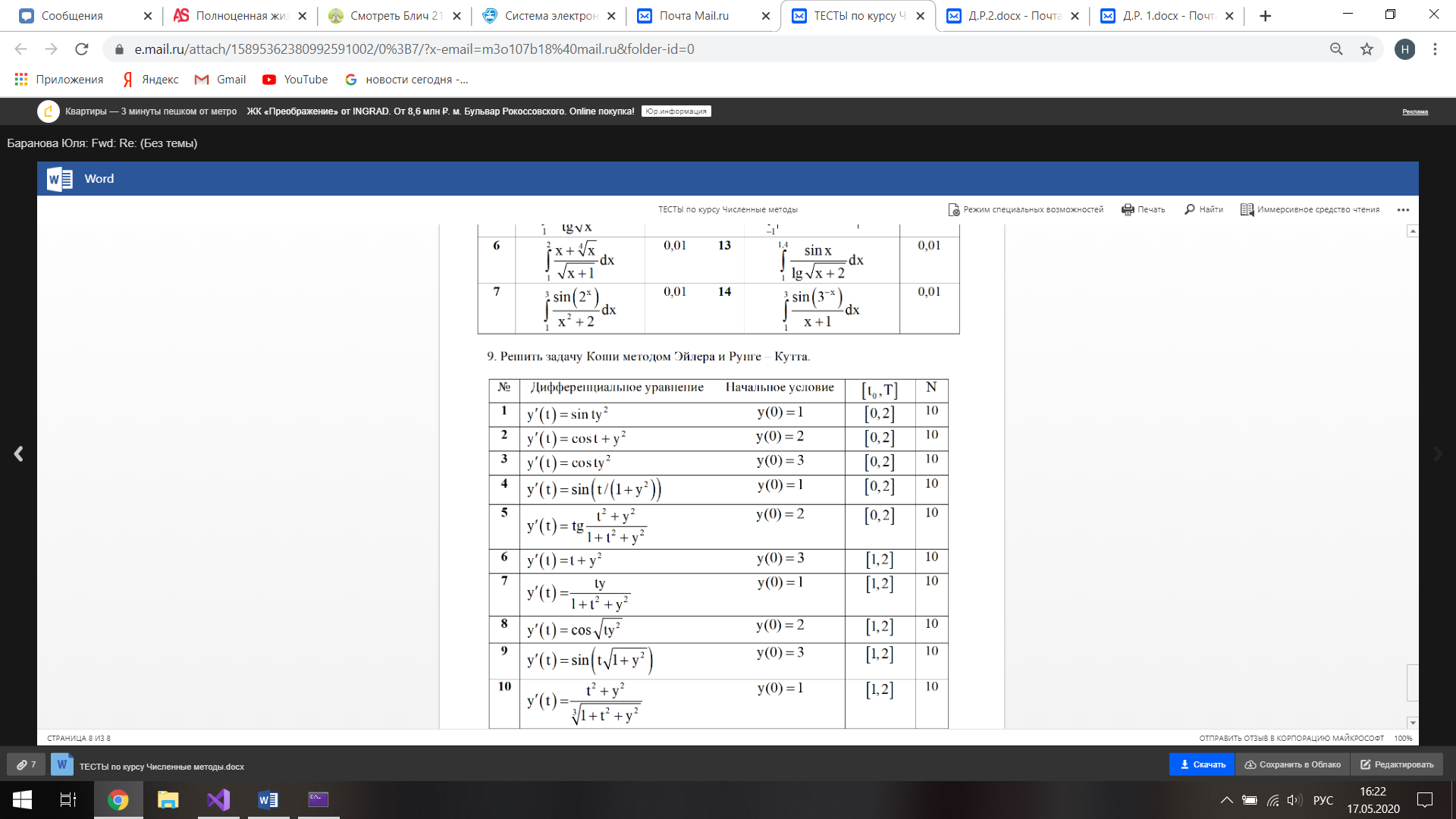
}

return l;

}

Результат работы программы





Части кода программы

double Fx25(double x, double y) { return cos(x) + y \* y; }

cout << "\n\t\t\t№ 9 " << endl;

КошиЭйлер(0, 2, 10, 0, Fx25);

void КошиЭйлер(double a, double b, int N, double y0, double (\*F)(double,double)) {

const int n = 6;

double NN[n];

char\*\* s;

s = new char\* [n];

for (int i = 0; i < n; i++)

s[i] = new char[15];

char s0[15] = { " i " };

char s1[15] = { " x " };

char s2[15] = { " f(x) " };

char s3[15] = { " y\_Эйдер " };

char s4[15] = { " y\_Рун-Кут\_2 " };

char s5[15] = { " y\_Рун-Кут\_4 " };

s[0] = s0;

s[1] = s1;

s[2] = s2;

s[3] = s3;

s[4] = s4;

s[5] = s5;

NN[0] = 0;

NN[1] = a;

NN[2] = F(a, y0);

NN[3] = y0;

NN[4] = y0;

NN[5] = y0;

double del;

double y = y0;

double t = y0;

double y2 = y0;

double y3 = y0;

double k1 = 0;

double k2 = 0;

double k3 = 0;

double k4 = 0;

del = (b - a) / double(N);

cout << "a = " << a << " b = " << b << " Шаг = " << del << " N = " << N << endl;

N\_tabl(n, s);

for (int j = 1; j <= N; j++)

{

C\_tabl(n, NN);

t = y;

y = t + del \* F(a + (j - 1) \* del, t);

y2 = t + del \* (F(a + (j - 1) \* del, t) + F(a + (j - 1) \* del, y))/2;

k1 = del \* F(a + (j - 1) \* del, y3);

k2 = del \* F(a + (j - 1) \* del + del / 2, y3 + k1 / 2);

k3 = del \* F(a + (j - 1) \* del + del / 2, y3 + k2 / 2);

k4 = del \* F(a + (j - 1) \* del + del, y3 + k3);

y3 = y3 + (k1 + 2 \* k2 + 2 \* k3 + k4)/6;

NN[0] = j;

NN[1] = a + j \* del;

NN[2] = F(a + j \* del, y);

NN[3] = y;

NN[4] = y2;

NN[5] = y3;

}

K\_tabl(n, NN);

}

Результат работы программы

