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

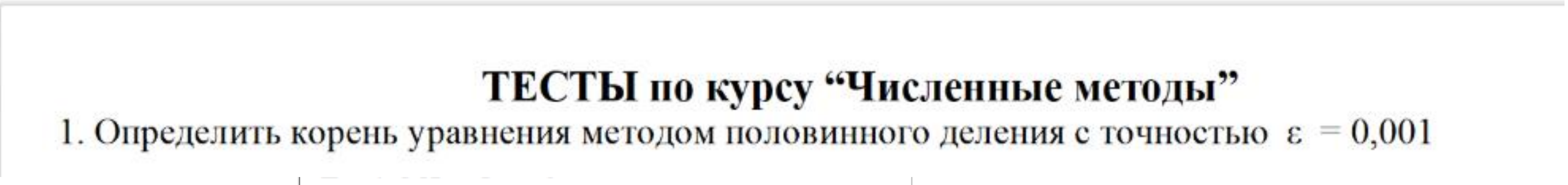
Выполнил:

Гордеев Н.М.

Принял:

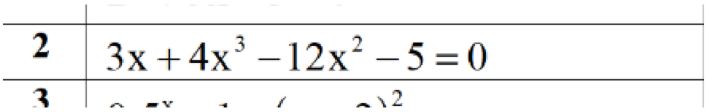
Яганов В.М.

МОСКВА 2020



ТЕСТЫ по курсу “Численные методы”

1. Определить корень уравнения методом половинного деления с точностью ε = 0,001

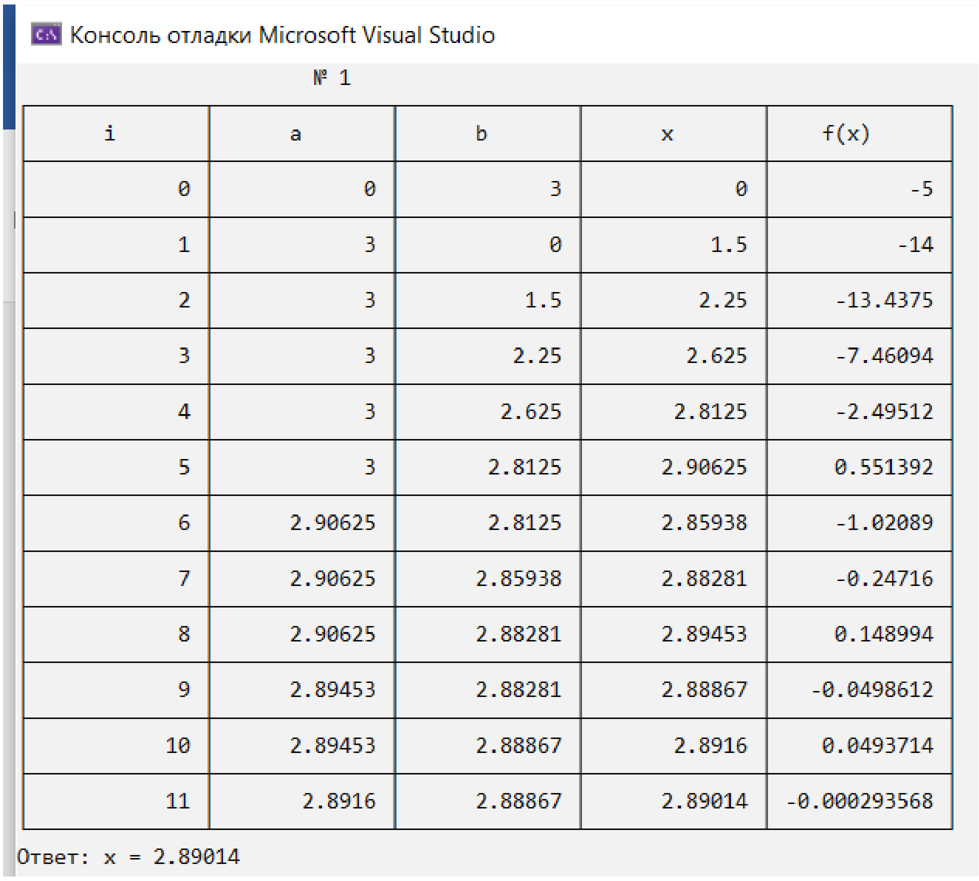


2	$3x+4x^3-12x^2-5=0$
3	

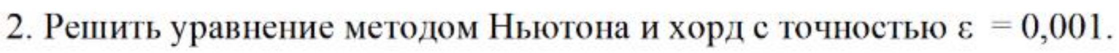
Части кода программы

```
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;
}
```

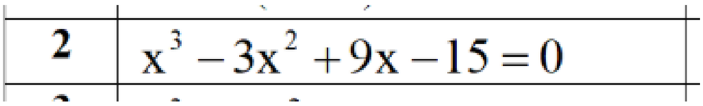
Результат работы программы



```
                      № 1
 _________________________________________________________________________
|      i      |      a      |      b      |      x      |     f(x)    |
|           0 |           0 |           3 |           0 |          -5 |
|           1 |           3 |           0 |         1.5 |         -14 |
|           2 |           3 |         1.5 |        2.25 |    -13.4375 |
|           3 |           3 |        2.25 |       2.625 |    -7.46094 |
|           4 |           3 |       2.625 |      2.8125 |    -2.49512 |
|           5 |           3 |      2.8125 |     2.90625 |    0.551392 |
|           6 |     2.90625 |      2.8125 |     2.85938 |    -1.02089 |
|           7 |     2.90625 |     2.85938 |     2.88281 |    -0.24716 |
|           8 |     2.90625 |     2.88281 |     2.89453 |    0.148994 |
|           9 |     2.89453 |     2.88281 |     2.88867 |  -0.0498612 |
|          10 |     2.89453 |     2.88867 |      2.8916 |   0.0493714 |
|          11 |      2.8916 |     2.88867 |     2.89014 |-0.000293568 |
Ответ: x = 2.89014
```



2. Решить уравнение методом Ньютона и хорд с точностью ε = 0,001.

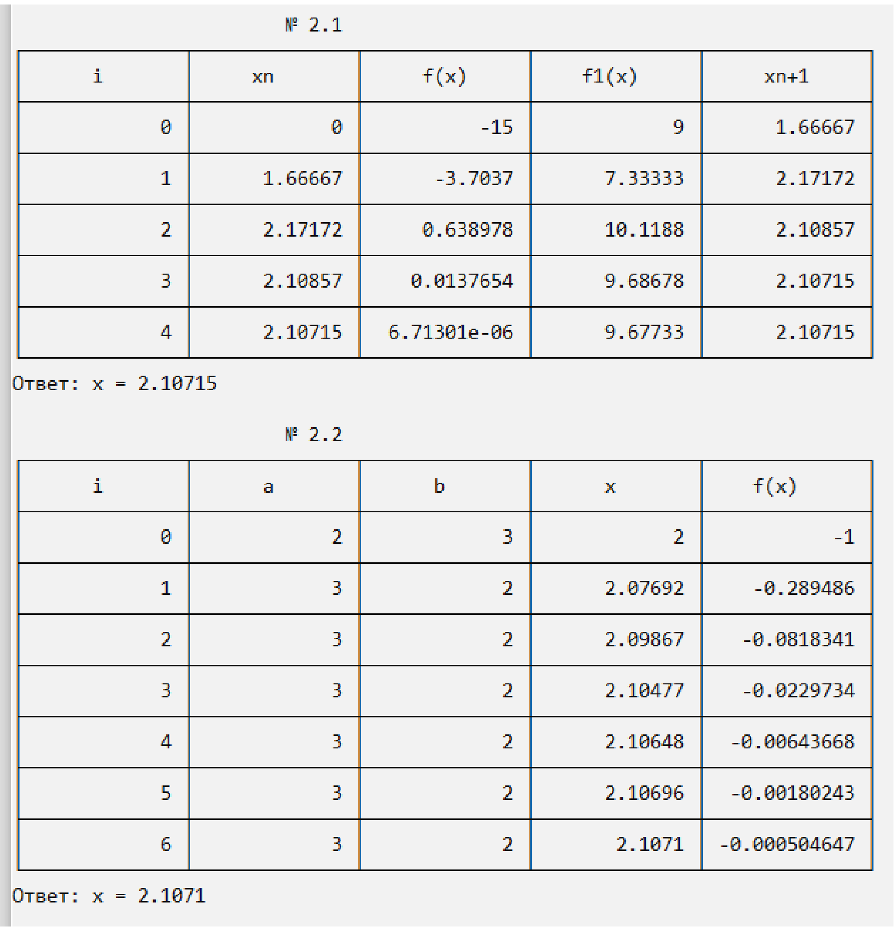


2	$x^3 - 3x^2 + 9x - 15 = 0$

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



№ 2.1

i	xn	f(x)	f1(x)	xn+1
0	0	-15	9	1.66667
1	1.66667	-3.7037	7.33333	2.17172
2	2.17172	0.638978	10.1188	2.10857
3	2.10857	0.0137654	9.68678	2.10715
4	2.10715	6.71301e-06	9.67733	2.10715

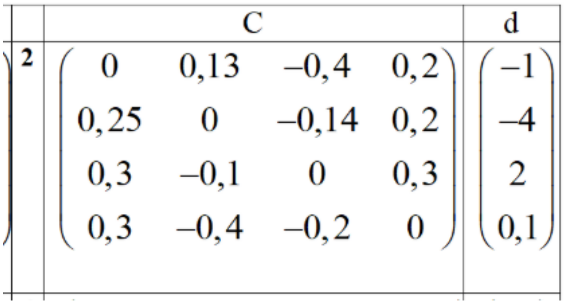
Ответ: x = 2.10715

№ 2.2

i	a	b	x	f(x)
0	2	3	2	-1
1	3	2	2.07692	-0.289486
2	3	2	2.09867	-0.0818341
3	3	2	2.10477	-0.0229734
4	3	2	2.10648	-0.00643668
5	3	2	2.10696	-0.00180243
6	3	2	2.1071	-0.000504647

Ответ: x = 2.1071

3. Решить систему $x = Cx + d$ методом простой итерации и Зейделя с точностью ε = 0,001



	C	d
2	$\begin{pmatrix} 0 & 0{,}13 & -0{,}4 & 0{,}2 \\ 0{,}25 & 0 & -0{,}14 & 0{,}2 \\ 0{,}3 & -0{,}1 & 0 & 0{,}3 \\ 0{,}3 & -0{,}4 & -0{,}2 & 0 \end{pmatrix}$	$\begin{pmatrix} -1 \\ -4 \\ 2 \\ 0{,}1 \end{pmatrix}$

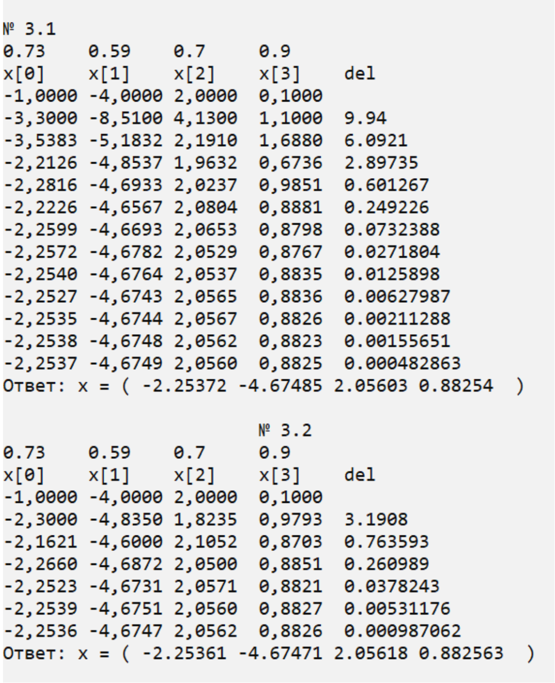
Части кода программы

```
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++)
```

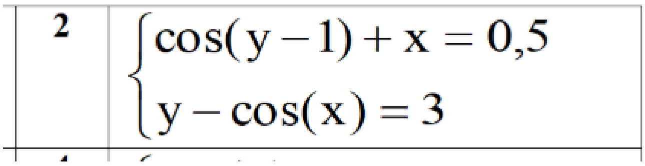
```
                for (int j = 0; j < m; j++)
                    f[i] += abs(c[j][i]);
        for (int i = 0; i < m; i++)
            if (f[i] < 1)
                k = 0;
        if (k)
            cout << "ошибка\n";
        cout << "\n";

        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



```
№ 3.1
0.73    0.59    0.7     0.9
x[0]    x[1]    x[2]    x[3]    del
-1,0000 -4,0000 2,0000  0,1000
-3,3000 -8,5100 4,1300  1,1000  9.94
-3,5383 -5,1832 2,1910  1,6880  6.0921
-2,2126 -4,8537 1,9632  0,6736  2.89735
-2,2816 -4,6933 2,0237  0,9851  0.601267
-2,2226 -4,6567 2,0804  0,8881  0.249226
-2,2599 -4,6693 2,0653  0,8798  0.0732388
-2,2572 -4,6782 2,0529  0,8767  0.0271804
-2,2540 -4,6764 2,0537  0,8835  0.0125898
-2,2527 -4,6743 2,0565  0,8836  0.00627987
-2,2535 -4,6744 2,0567  0,8826  0.00211288
-2,2538 -4,6748 2,0562  0,8823  0.00155651
-2,2537 -4,6749 2,0560  0,8825  0.000482863
Ответ: x = ( -2.25372 -4.67485 2.05603 0.88254   )

                        № 3.2
0.73    0.59    0.7     0.9
x[0]    x[1]    x[2]    x[3]    del
-1,0000 -4,0000 2,0000  0,1000
-2,3000 -4,8350 1,8235  0,9793  3.1908
-2,1621 -4,6000 2,1052  0,8703  0.763593
-2,2660 -4,6872 2,0500  0,8851  0.260989
-2,2523 -4,6731 2,0571  0,8821  0.0378243
-2,2539 -4,6751 2,0560  0,8827  0.00531176
-2,2536 -4,6747 2,0562  0,8826  0.000987062
Ответ: x = ( -2.25361 -4.67471 2.05618 0.882563   )
```

4. Решить систему методом простой итерации и Ньютона с точностью ε = 0,001.



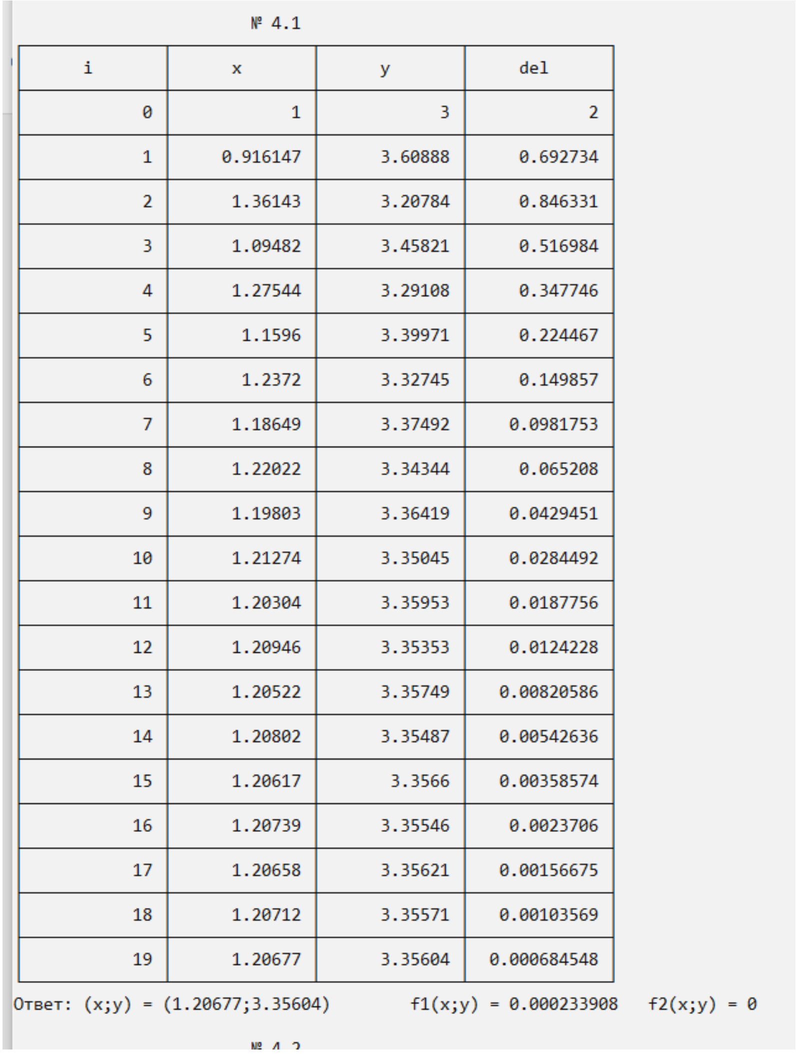
2	$\begin{cases}\cos(y-1)+x=0{,}5\\ y-\cos(x)=3\end{cases}$

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 &x, double &y, 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] = x;
	N[2] = y;

	double x = x;
	double y = y;
	double t1 = x-1;
	double t2 = y-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);
        x = x; y = y;
}

void Ньютон(double& x, double& y, 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] = x;
        N[2] = y;
        N[3] = F1(x,y);
        N[4] = F2(x, y);
        N[5] = F3(x, y);

        double x = x;
        double y = y;
        double t1 = x - 1;
        double t2 = y - 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);
        x = x; y = y;
    }
```

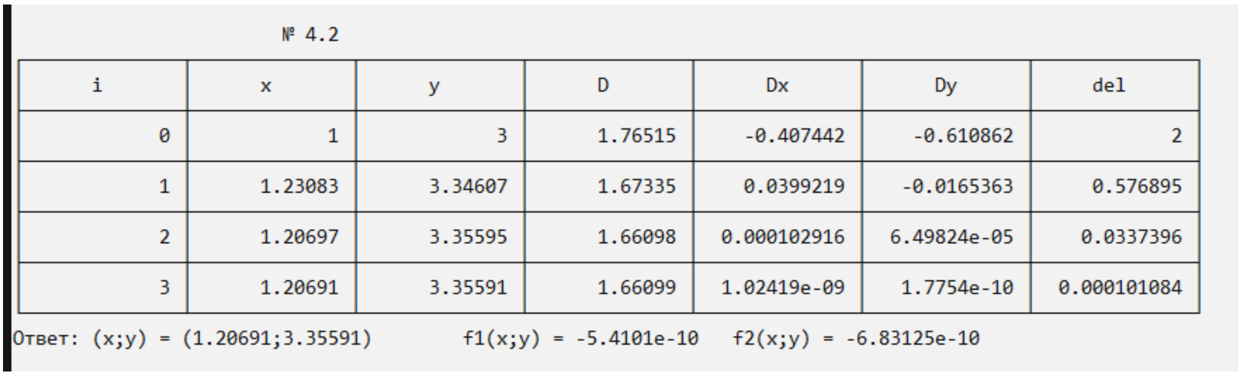
Результат работы программы



№ 4.1

i	x	y	del
0	1	3	2
1	0.916147	3.60888	0.692734
2	1.36143	3.20784	0.846331
3	1.09482	3.45821	0.516984
4	1.27544	3.29108	0.347746
5	1.1596	3.39971	0.224467
6	1.2372	3.32745	0.149857
7	1.18649	3.37492	0.0981753
8	1.22022	3.34344	0.065208
9	1.19803	3.36419	0.0429451
10	1.21274	3.35045	0.0284492
11	1.20304	3.35953	0.0187756
12	1.20946	3.35353	0.0124228
13	1.20522	3.35749	0.00820586
14	1.20802	3.35487	0.00542636
15	1.20617	3.3566	0.00358574
16	1.20739	3.35546	0.0023706
17	1.20658	3.35621	0.00156675
18	1.20712	3.35571	0.00103569
19	1.20677	3.35604	0.000684548

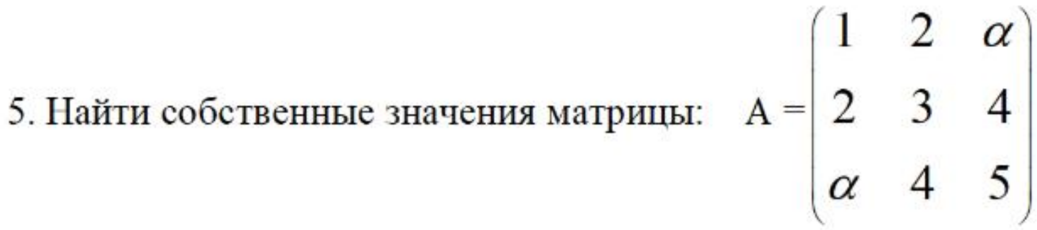
```
Ответ: (x;y) = (1.20677;3.35604)        f1(x;y) = 0.000233908    f2(x;y) = 0
```



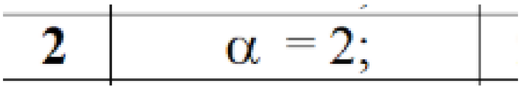
№ 4.2

i	x	y	D	Dx	Dy	del
0	1	3	1.76515	-0.407442	-0.610862	2
1	1.23083	3.34607	1.67335	0.0399219	-0.0165363	0.576895
2	1.20697	3.35595	1.66098	0.000102916	6.49824e-05	0.0337396
3	1.20691	3.35591	1.66099	1.02419e-09	1.7754e-10	0.000101084

```
Ответ: (x;y) = (1.20691;3.35591)        f1(x;y) = -5.4101e-10    f2(x;y) = -6.83125e-10
```



5. Найти собственные значения матрицы: $A = \begin{pmatrix} 1 & 2 & \alpha \\ 2 & 3 & 4 \\ \alpha & 4 & 5 \end{pmatrix}$

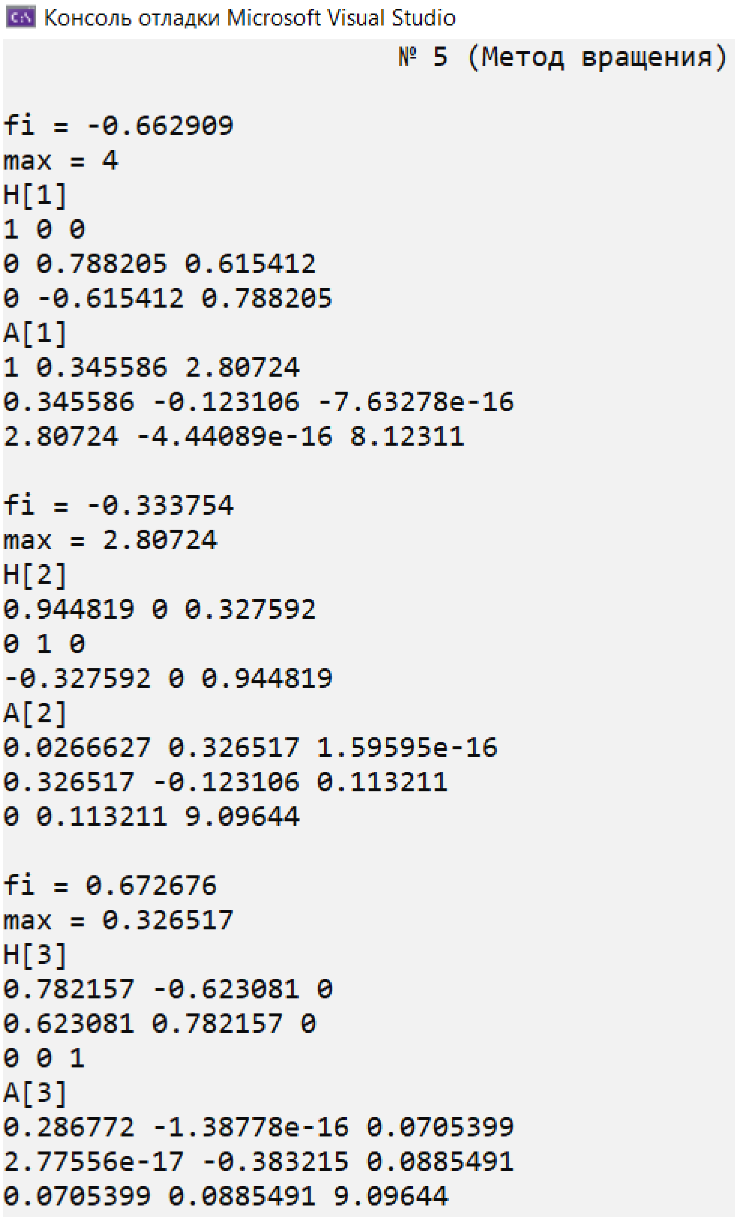


2	$\alpha = 2;$

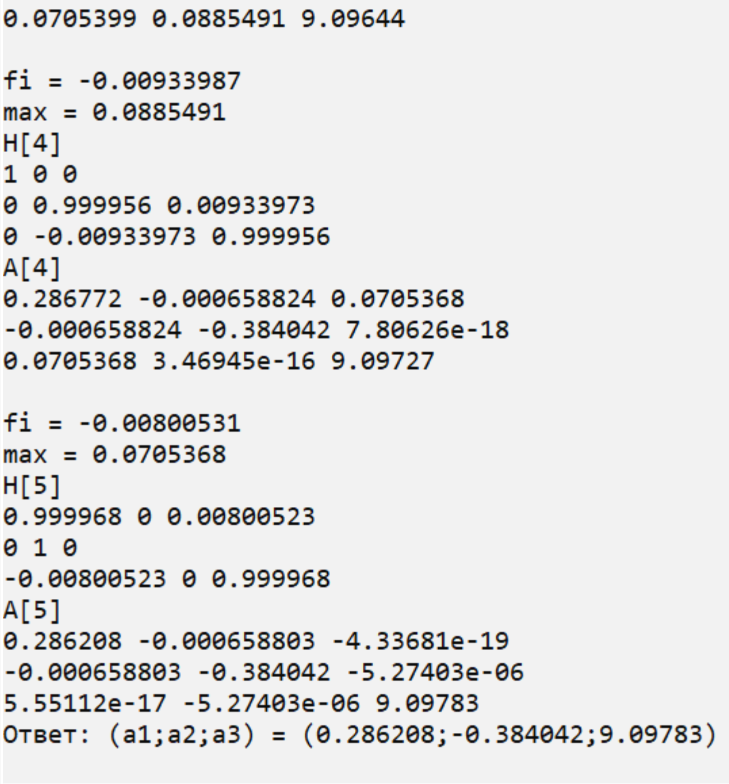
Части кода программы

```
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];
}
```

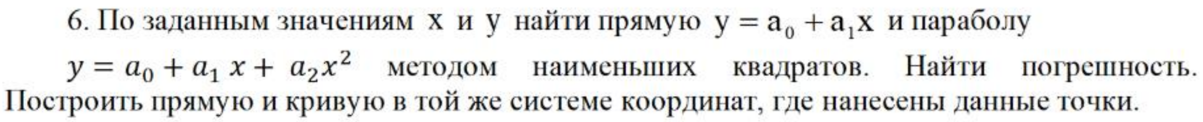
Результат работы программы



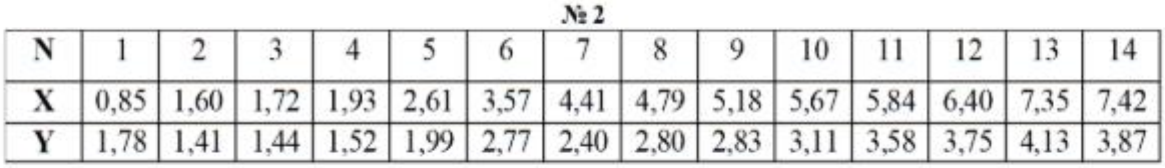
```
                                                № 5 (Метод вращения)

fi = -0.662909
max = 4
H[1]
1 0 0
0 0.788205 0.615412
0 -0.615412 0.788205
A[1]
1 0.345586 2.80724
0.345586 -0.123106 -7.63278e-16
2.80724 -4.44089e-16 8.12311

fi = -0.333754
max = 2.80724
H[2]
0.944819 0 0.327592
0 1 0
-0.327592 0 0.944819
A[2]
0.0266627 0.326517 1.59595e-16
0.326517 -0.123106 0.113211
0 0.113211 9.09644

fi = 0.672676
max = 0.326517
H[3]
0.782157 -0.623081 0
0.623081 0.782157 0
0 0 1
A[3]
0.286772 -1.38778e-16 0.0705399
2.77556e-17 -0.383215 0.0885491
0.0705399 0.0885491 9.09644
```



```
0.0705399 0.0885491 9.09644

fi = -0.00933987
max = 0.0885491
H[4]
1 0 0
0 0.999956 0.00933973
0 -0.00933973 0.999956
A[4]
0.286772 -0.000658824 0.0705368
-0.000658824 -0.384042 7.80626e-18
0.0705368 3.46945e-16 9.09727

fi = -0.00800531
max = 0.0705368
H[5]
0.999968 0 0.00800523
0 1 0
-0.00800523 0 0.999968
A[5]
0.286208 -0.000658803 -4.33681e-19
-0.000658803 -0.384042 -5.27403e-06
5.55112e-17 -5.27403e-06 9.09783
Ответ: (a1;a2;a3) = (0.286208;-0.384042;9.09783)
```



6. По заданным значениям x и y найти прямую $y = a_0 + a_1x$ и параболу $y = a_0 + a_1\,x + a_2x^2$ методом наименьших квадратов. Найти погрешность. Построить прямую и кривую в той же системе координат, где нанесены данные точки.



№ 2

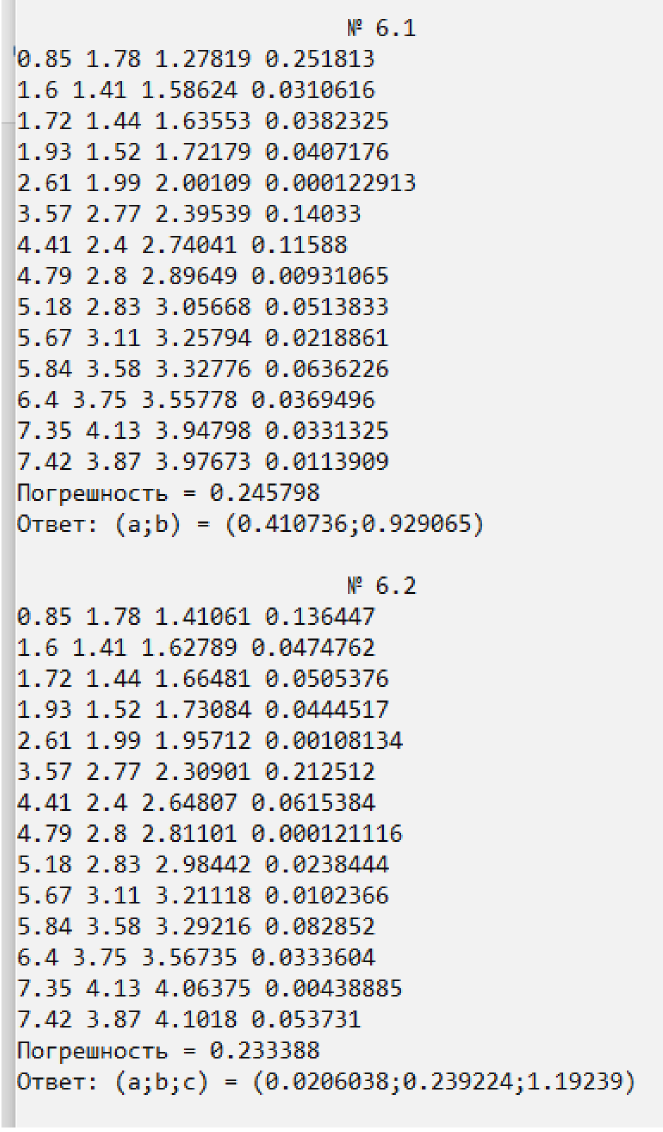
N	1	2	3	4	5	6	7	8	9	10	11	12	13	14
X	0,85	1,60	1,72	1,93	2,61	3,57	4,41	4,79	5,18	5,67	5,84	6,40	7,35	7,42
Y	1,78	1,41	1,44	1,52	1,99	2,77	2,40	2,80	2,83	3,11	3,58	3,75	4,13	3,87

Части кода программы

```
double x[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 y[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(x, y, ab, 14);
	cout << "Ответ: (a;b) = (" << ab[1] << ";" << ab[0] << ")" << endl;

	cout << "\n\t\t\t№ 6.2 " << endl;
	НаимКвЛин(x, y, ab, 14,0);
	cout << "Ответ: (a;b;c) = (" << ab[2] << ";" << ab[1] << ";" << ab[0] << ")" << 
endl;

void НаимКвЛин(double* x, double* y, 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 += x[i];
		x2 += x[i] * x[i];
		x3 += x[i] * x[i] * x[i];
		x4 += x[i] * x[i] * x[i] * x[i];
		y += y[i];
		xy += y[i] * x[i];
		x2y += y[i] * x[i] * x[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(x[i] * a[1] + a[0] - y[i],2);
			cout << x[i] << " " << y[i] << " " << x[i] * a[1] + a[0] << " " << 
pow(x[i] * a[1] + a[0] - y[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);
```

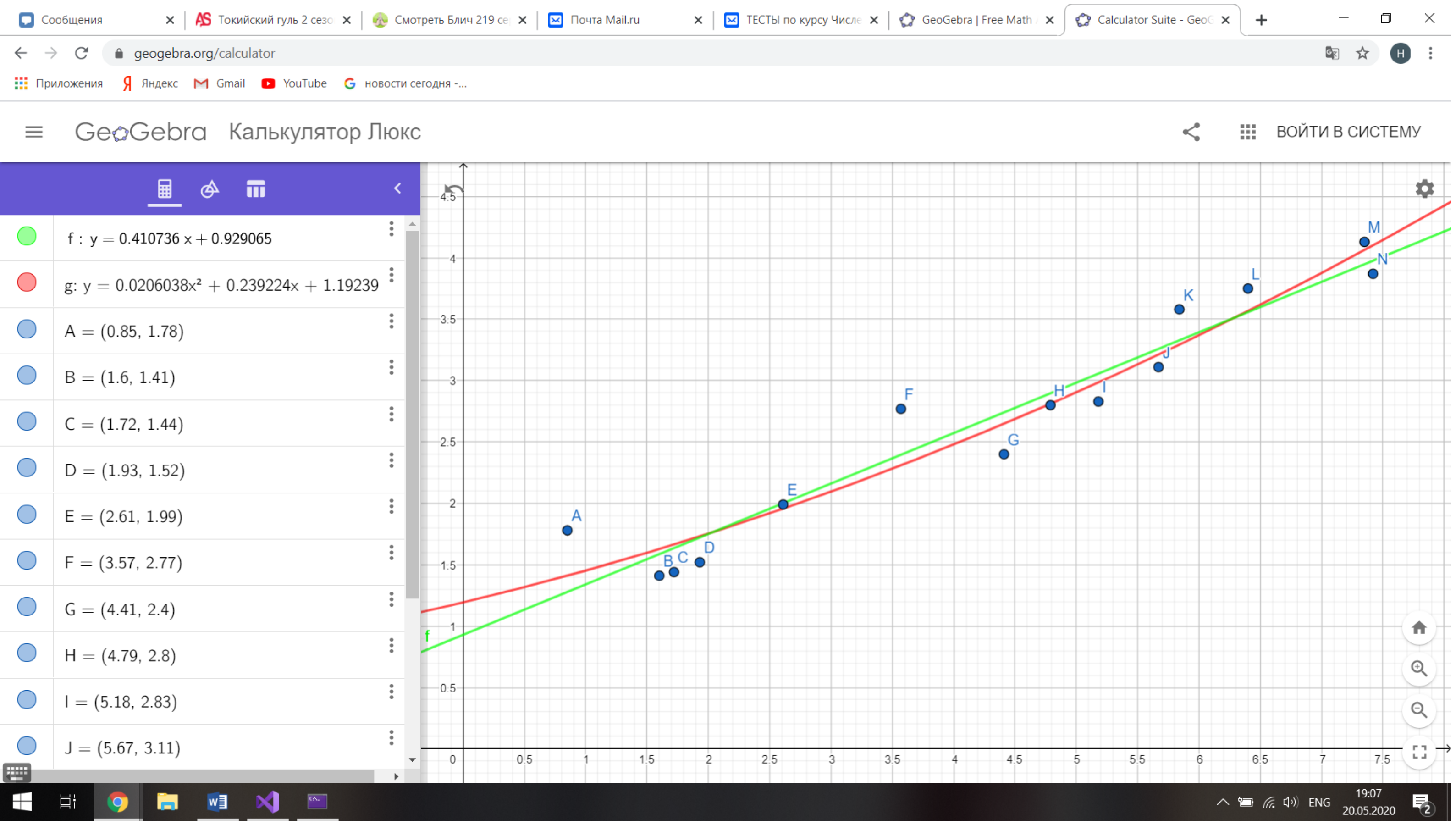
```
			a[2] = (k * x2 * x2y + x * xy * x2 + x * x3 * y - y * x2 * x2 - x * x * x2y
- xy * x3 * k) / (k * x2 * x4 + x * x3 * x2 + x * x3 * x2 - x2 * x2 * x2 - x * x * x4 -
x3 * x3 * k);
			for (size_t i = 0; i < k; i++)
			{
				t1 += pow(x[i] * x[i] * a[2] + x[i] * a[1] + a[0] - y[i],2);
				cout << x[i] << " " << y[i] << " " << x[i] * x[i] * a[2] + x[i] *
a[1] + a[0] << " " << pow(x[i] * x[i] * a[2] + x[i] * a[1] + a[0] - y[i], 2) << "\n";
			}
		}
		cout << "Погрешность = "<< pow((t1/k),0.5) <<"\n";
}
```

Результат работы программы



```
                        № 6.1
0.85 1.78 1.27819 0.251813
1.6 1.41 1.58624 0.0310616
1.72 1.44 1.63553 0.0382325
1.93 1.52 1.72179 0.0407176
2.61 1.99 2.00109 0.000122913
3.57 2.77 2.39539 0.14033
4.41 2.4 2.74041 0.11588
4.79 2.8 2.89649 0.00931065
5.18 2.83 3.05668 0.0513833
5.67 3.11 3.25794 0.0218861
5.84 3.58 3.32776 0.0636226
6.4 3.75 3.55778 0.0369496
7.35 4.13 3.94798 0.0331325
7.42 3.87 3.97673 0.0113909
Погрешность = 0.245798
Ответ: (a;b) = (0.410736;0.929065)

                        № 6.2
0.85 1.78 1.41061 0.136447
1.6 1.41 1.62789 0.0474762
1.72 1.44 1.66481 0.0505376
1.93 1.52 1.73084 0.0444517
2.61 1.99 1.95712 0.00108134
3.57 2.77 2.30901 0.212512
4.41 2.4 2.64807 0.0615384
4.79 2.8 2.81101 0.000121116
5.18 2.83 2.98442 0.0238444
5.67 3.11 3.21118 0.0102366
5.84 3.58 3.29216 0.082852
6.4 3.75 3.56735 0.0333604
7.35 4.13 4.06375 0.00438885
7.42 3.87 4.1018 0.053731
Погрешность = 0.233388
Ответ: (a;b;c) = (0.0206038;0.239224;1.19239)
```

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



f : y = 0.410736 x + 0.929065

g: y = 0.0206038x² + 0.239224x + 1.19239

A = (0.85, 1.78)

B = (1.6, 1.41)

C = (1.72, 1.44)

D = (1.93, 1.52)

E = (2.61, 1.99)

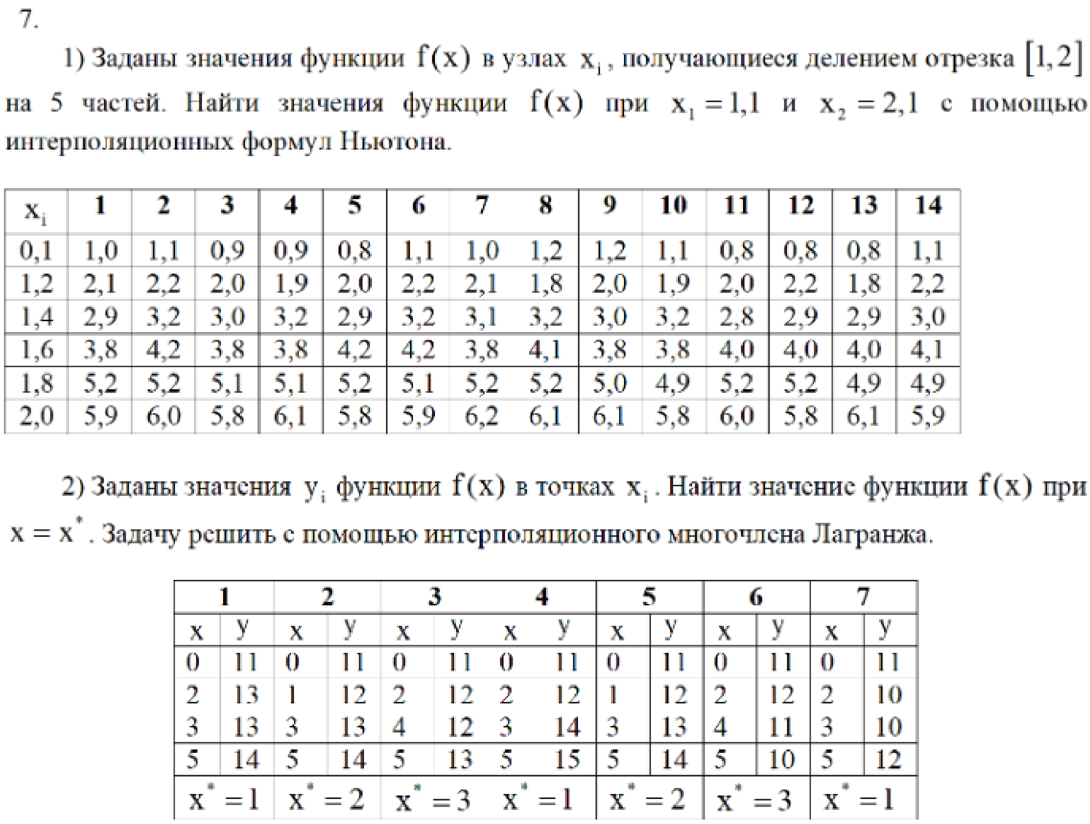
F = (3.57, 2.77)

G = (4.41, 2.4)

H = (4.79, 2.8)

I = (5.18, 2.83)

J = (5.67, 3.11)



7.

1) Заданы значения функции $f(x)$ в узлах x_i, получающиеся делением отрезка $[1,2]$ на 5 частей. Найти значения функции $f(x)$ при $x_1 = 1{,}1$ и $x_2 = 2{,}1$ с помощью интерполяционных формул Ньютона.

x_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0,1	1,0	1,1	0,9	0,9	0,8	1,1	1,0	1,2	1,2	1,1	0,8	0,8	0,8	1,1
1,2	2,1	2,2	2,0	1,9	2,0	2,2	2,1	1,8	2,0	1,9	2,0	2,2	1,8	2,2
1,4	2,9	3,2	3,0	3,2	2,9	3,2	3,1	3,2	3,0	3,2	2,8	2,9	2,9	3,0
1,6	3,8	4,2	3,8	3,8	4,2	4,2	3,8	4,1	3,8	3,8	4,0	4,0	4,0	4,1
1,8	5,2	5,2	5,1	5,1	5,2	5,1	5,2	5,2	5,0	4,9	5,2	5,2	4,9	4,9
2,0	5,9	6,0	5,8	6,1	5,8	5,9	6,2	6,1	6,1	5,8	6,0	5,8	6,1	5,9

2) Заданы значения y_i функции $f(x)$ в точках x_i. Найти значение функции $f(x)$ при $x = x^*$. Задачу решить с помощью интерполяционного многочлена Лагранжа.

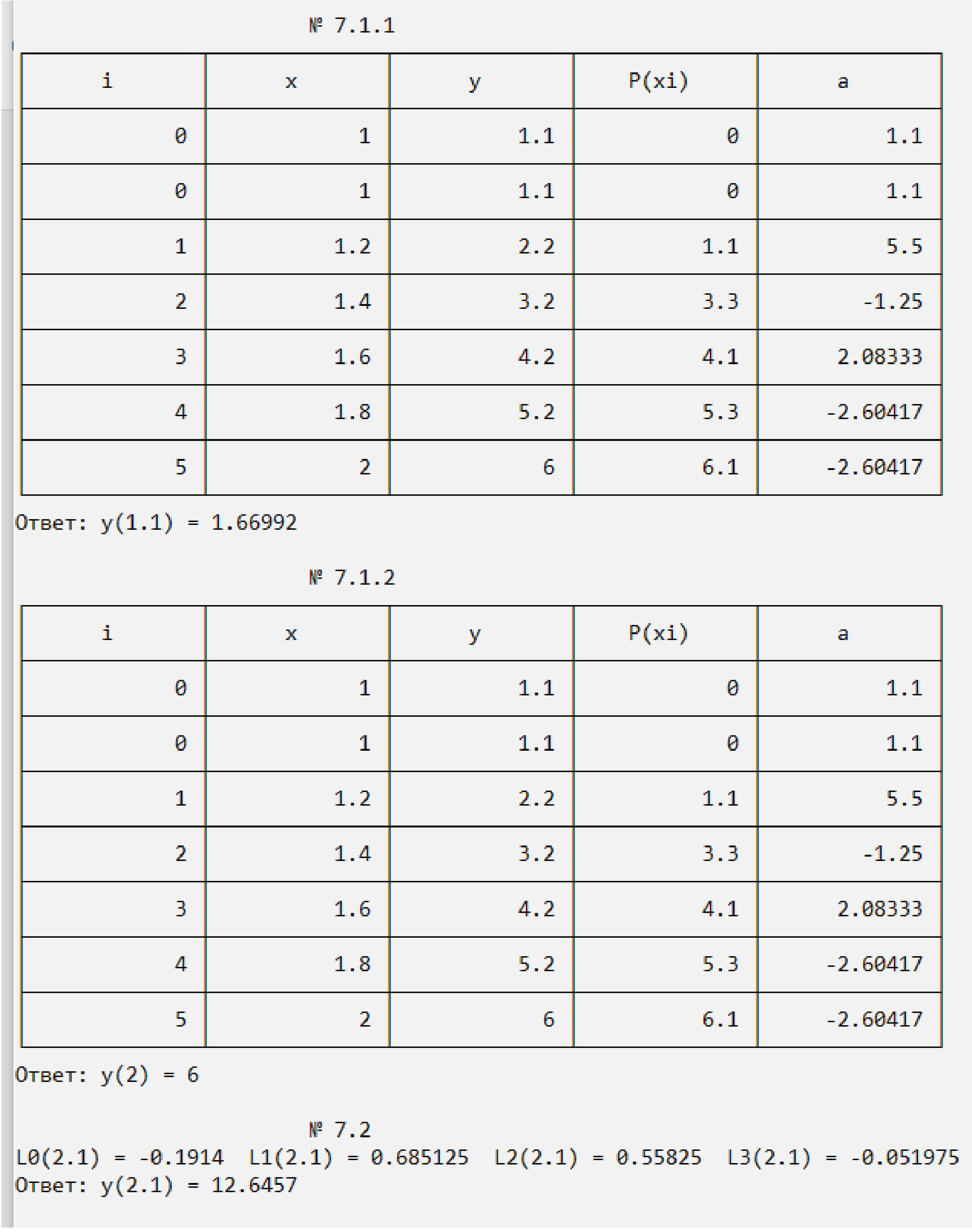
1		2		3		4		5		6		7	
x	y	x	y	x	y	x	y	x	y	x	y	x	y
0	11	0	11	0	11	0	11	0	11	0	11	0	11
2	13	1	12	2	12	2	12	1	12	2	12	2	10
3	13	3	13	4	12	3	14	3	13	4	11	3	10
5	14	5	14	5	13	5	15	5	14	5	10	5	12
$x^* = 1$		$x^* = 2$		$x^* = 3$		$x^* = 1$		$x^* = 2$		$x^* = 3$		$x^* = 1$	

Части кода программы

```
const int m2 = 4;
double x1[m2] = { 0, 1, 3, 5 };
double y1[m2] = { 11,12,13,14 };
const int m3 = 6;
double x2[m3] = { 1,  1.2, 1.4, 1.6, 1.8, 2 };
double y2[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(x2, y2, ab2, m3, 1.1) << endl;

    cout << "\n\t\t\t№ 7.1.2 " << endl;
    cout << "Ответ: y(2) = " << ИнтерполяцияНьютон(x2, y2, ab2, m3, 2) << endl;

    double ab3[m2];
    cout << "\n\t\t\t№ 7.2 " << endl;
    cout << "Ответ: y(2.1) = " << ИнтерполяцияЛогранж(x1, y1, 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



№ 7.1.1

i	x	y	P(xi)	a
0	1	1.1	0	1.1
0	1	1.1	0	1.1
1	1.2	2.2	1.1	5.5
2	1.4	3.2	3.3	-1.25
3	1.6	4.2	4.1	2.08333
4	1.8	5.2	5.3	-2.60417
5	2	6	6.1	-2.60417

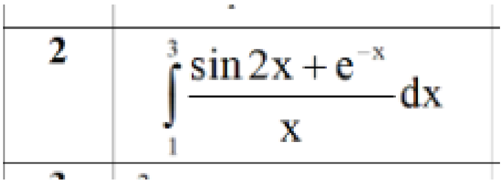
```
Ответ: y(1.1) = 1.66992
```

№ 7.1.2

i	x	y	P(xi)	a
0	1	1.1	0	1.1
0	1	1.1	0	1.1
1	1.2	2.2	1.1	5.5
2	1.4	3.2	3.3	-1.25
3	1.6	4.2	4.1	2.08333
4	1.8	5.2	5.3	-2.60417
5	2	6	6.1	-2.60417

```
Ответ: y(2) = 6

                        № 7.2
L0(2.1) = -0.1914  L1(2.1) = 0.685125  L2(2.1) = 0.55825  L3(2.1) = -0.051975
Ответ: y(2.1) = 12.6457
```

8. Вычислить определённый интеграл с точностью $\varepsilon = 0{,}01$ методом Симпсона.

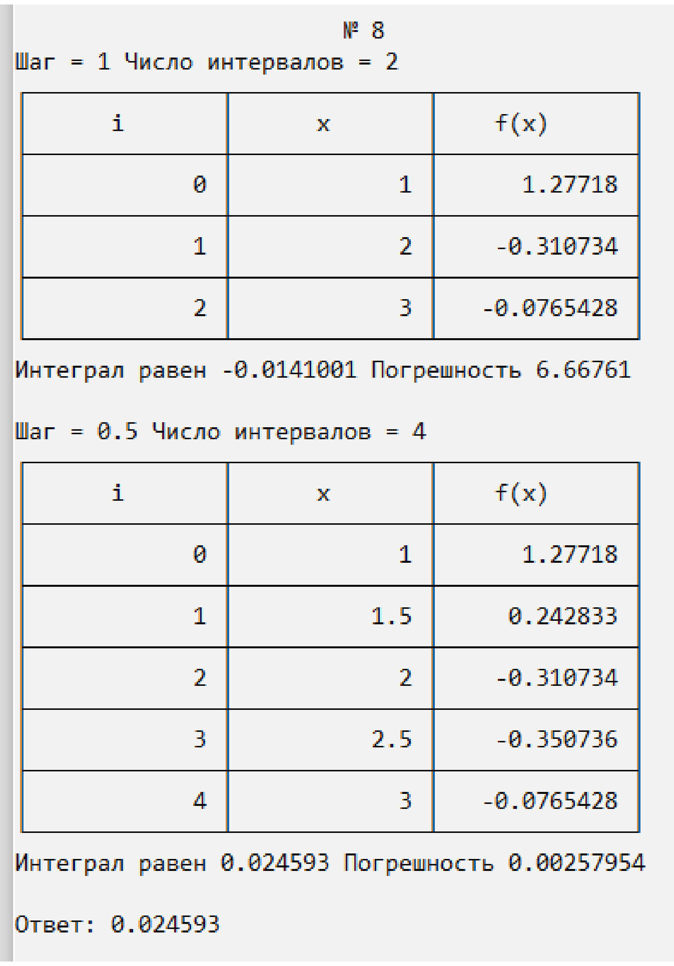


2	$\int_1^3 \frac{\sin 2x + e^{-x}}{x} dx$

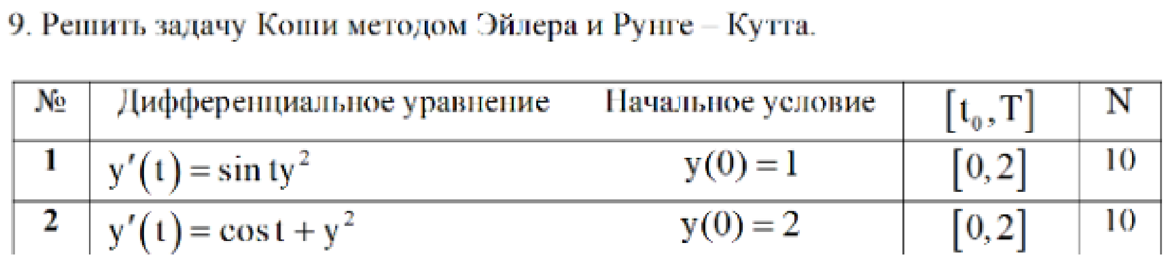
Части кода программы

```
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



```
                        № 8
Шаг = 1 Число интервалов = 2
┌──────────────┬──────────────┬──────────────┐
│      i       │      x       │     f(x)     │
├──────────────┼──────────────┼──────────────┤
│            0 │            1 │      1.27718 │
├──────────────┼──────────────┼──────────────┤
│            1 │            2 │    -0.310734 │
├──────────────┼──────────────┼──────────────┤
│            2 │            3 │   -0.0765428 │
└──────────────┴──────────────┴──────────────┘
Интеграл равен -0.0141001 Погрешность 6.66761

Шаг = 0.5 Число интервалов = 4
┌──────────────┬──────────────┬──────────────┐
│      i       │      x       │     f(x)     │
├──────────────┼──────────────┼──────────────┤
│            0 │            1 │      1.27718 │
├──────────────┼──────────────┼──────────────┤
│            1 │          1.5 │     0.242833 │
├──────────────┼──────────────┼──────────────┤
│            2 │            2 │    -0.310734 │
├──────────────┼──────────────┼──────────────┤
│            3 │          2.5 │    -0.350736 │
├──────────────┼──────────────┼──────────────┤
│            4 │            3 │   -0.0765428 │
└──────────────┴──────────────┴──────────────┘
Интеграл равен 0.024593 Погрешность 0.00257954

Ответ: 0.024593
```



9. Решить задачу Коши методом Эйлера и Рунге – Кутта.

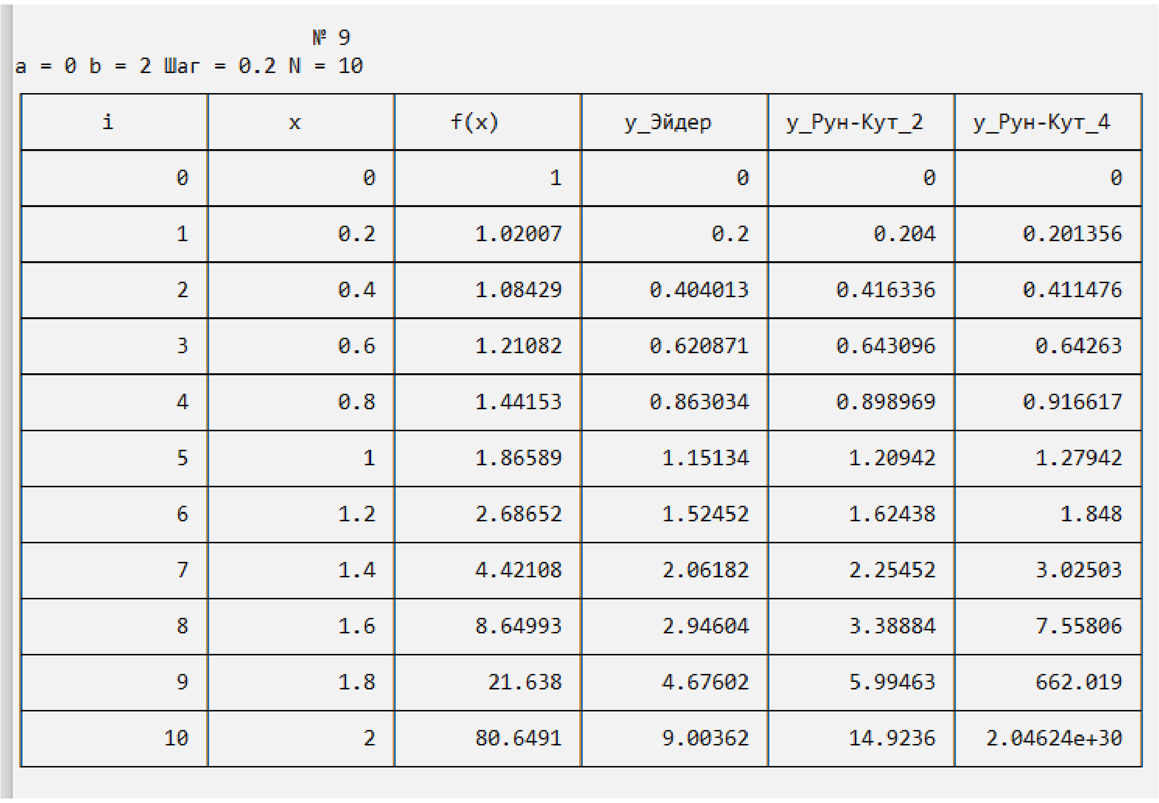
№	Дифференциальное уравнение	Начальное условие	$[t_0, T]$	N
1	$y'(t) = \sin ty^2$	$y(0) = 1$	$[0,2]$	10
2	$y'(t) = \cos t + y^2$	$y(0) = 2$	$[0,2]$	10

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] = { "   у_Эйдер    " };
       char s4[15] = { " у_Рун-Кут_2  " };
       char s5[15] = { " у_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



```
                           № 9
a = 0 b = 2 Шаг = 0.2 N = 10

      i           x          f(x)       y_Эйдер   y_Рун-Кут_2  y_Рун-Кут_4

          0           0           1           0           0            0
          1         0.2     1.02007         0.2       0.204     0.201356
          2         0.4     1.08429    0.404013    0.416336     0.411476
          3         0.6     1.21082    0.620871    0.643096      0.64263
          4         0.8     1.44153    0.863034    0.898969     0.916617
          5           1     1.86589     1.15134     1.20942      1.27942
          6         1.2     2.68652     1.52452     1.62438        1.848
          7         1.4     4.42108     2.06182     2.25452      3.02503
          8         1.6     8.64993     2.94604     3.38884      7.55806
          9         1.8      21.638     4.67602     5.99463      662.019
         10           2     80.6491     9.00362     14.9236  2.04624e+30
```