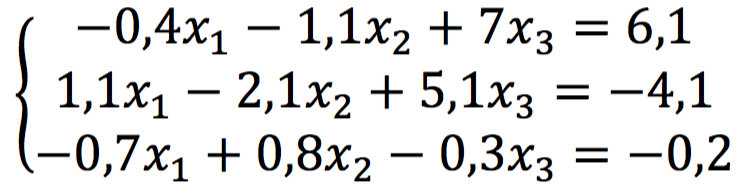
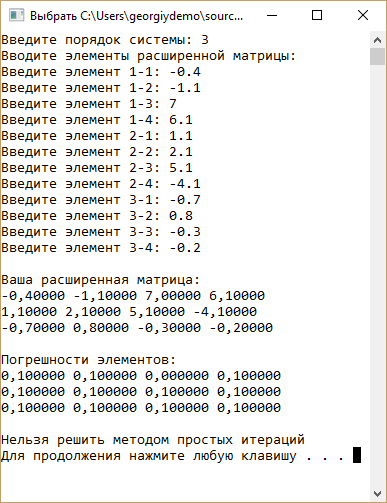
Условие:

Выполнить программную реализацию метода простой итерации для решения системы линейных уравнений с точностью до пятого знака после запятой:

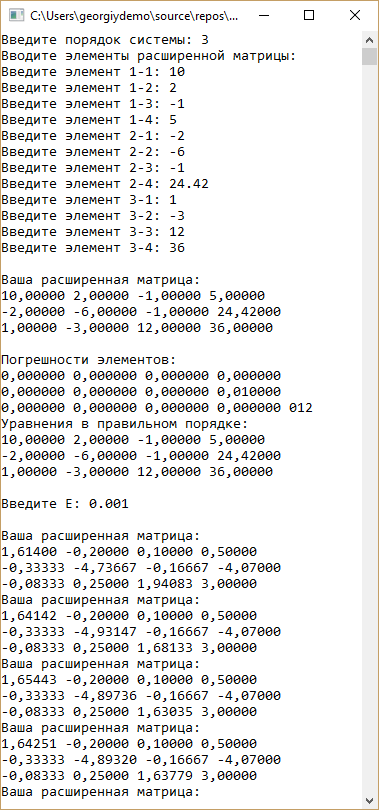
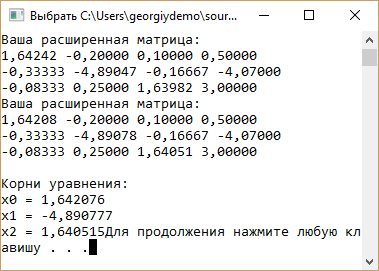


Исходный код программы:  
  
#include **<stdlib.h>**#include **<stdio.h>**#include **<math.h>**#include **<locale.h>**#include **<iostream>  
using namespace** std;  
  
**void** change(**double**\*\* nc, **int** i1, **int** i2, **int** n);  
  
**int** main()  
{  
 **double** \*\*data, k, help, \*helper, \*\*miss, \*mishelp, copy, E, \*h2;  
 **int** i, j, l, \*hlp, n, counter;  
   
 printf(**"Введите порядок системы: "**);  
 cin >> n;  
 data = (**double**\*\*)malloc(n \* **sizeof**(**double**\*));  
 miss = (**double**\*\*)malloc(n \* **sizeof**(**double**\*));  
 hlp = (**int**\*)malloc(n \* **sizeof**(**int**));  
 helper = (**double**\*)malloc(n \* **sizeof**(**int**));  
 h2 = (**double**\*)malloc(n \* **sizeof**(**int**));  
 **for** (i = 0; i < n; i++) {  
 data[i] = (**double**\*)malloc((n + 1) \* **sizeof**(**double**));  
 miss[i] = (**double**\*)malloc((n + 1) \* **sizeof**(**double**));  
 };  
 printf(**"Вводите элементы расширенной матрицы: \n"**);  
 **for** (i = 0; i < n; i++)  
 **for** (j = 0; j < n + 1; j++) {  
 printf(**"Введите элемент %d-%d: "**, i + 1, j + 1);  
 cin >> data[i][j];  
 miss[i][j] = 1;  
 copy = data[i][j];  
 **while** (floor(copy) != ceil(copy)) {  
 copy \*= 10;  
 miss[i][j] /= 10;  
 *//printf("+");* }  
 **if** ((data[i][j] - floor(data[i][j]) >= 0.001) && (data[i][j] - floor(data[i][j]) <= 0.009))  
 miss[i][j] = 0.001;  
 **if** (miss[i][j] == 1)  
 miss[i][j] = 0;  
 }  
 printf(**"\nВаша расширенная матрица: "**);  
 **for** (i = 0; i < n; i++) {  
 printf(**"\n"**);  
 **for** (j = 0; j < n + 1; j++)  
 printf(**"%5.5lf "**, data[i][j]);  
 }  
 printf(**"\n\nПогрешности элементов: "**);  
 **for** (i = 0; i < n; i++) {  
 printf(**"\n"**);  
 **for** (j = 0; j < n + 1; j++)  
 printf(**"%lf "**, miss[i][j]);  
 }  
 **for** (i = 0; i < n; i++) {  
 hlp[i] = 0;  
 help = fabs(data[i][0]);  
 copy = help;  
 **for** (j = 1; j < n; j++) {  
 copy = copy + fabs(data[i][j]);  
 **if** (fabs(data[i][j]) > help) {  
 help = fabs(data[i][j]);  
 hlp[i] = j;  
 }  
 }  
 **if** (fabs(copy - fabs(help)) > fabs(help)) {  
 cout << **"\n\nНельзя решить методом Простых итераций!"**;  
 system(**"pause"**);  
 exit(0);  
 }  
 }  
 **for** (i = 0; i < n; i++)  
 printf(**"%d"**, hlp[i]);  
 **for** (i = 0; i < n; i++)  
 **for** (j = i + 1; j < n; j++)  
 **if** (hlp[j] == hlp[i]) {  
 printf(**"\n\nНельзя решить методом Простых итераций\n "**);  
 system(**"pause"**);  
 exit(0);  
 }  
 **for** (i = 0; i < n; i++) {  
 **for** (j = i; j < n; j++)  
 **if** (hlp[j] == i) {  
 change(data, i, j, n);  
 change(miss, i, j, n);  
 help = hlp[i];  
 hlp[i] = hlp[j];  
 hlp[j] = help;  
 **break**;  
 }  
 }  
 printf(**"\nУравнения в правильном порядке: "**);  
 **for** (i = 0; i < n; i++) {  
 printf(**"\n"**);  
 **for** (j = 0; j < n + 1; j++)  
 printf(**"%5.5lf "**, data[i][j]);  
 }  
 **for** (i = 0; i < n; i++) {  
 data[i][n] = data[i][n] / data[i][i];  
 miss[i][n] = (fabs(data[i][n]) \* miss[i][i] + fabs(data[i][i]) \* miss[i][n]) / (data[i][i] \* data[i][i]);  
 **for** (j = n - 1; j >= 0; j--)  
  
 {  
 **if** (j != i) {  
 miss[i][j] = (fabs(data[i][j]) \* miss[i][i] + fabs(data[i][i]) \* miss[i][j]) / (data[i][i] \* data[i][i]);  
 data[i][j] = -data[i][j] / data[i][i];  
 }  
 }  
 data[i][i] = data[i][n];  
 miss[i][i] = miss[i][n];  
 }  
 printf(**"\n\nВведите E: "**);  
 cin >> E;  
 **do** {  
 **for** (i = 0; i < n; i++) {  
 helper[i] = data[i][i]; *//коэффициенты при главных элементах* h2[i] = helper[i];  
 data[i][i] = data[i][n]; *//текущий х1, х2 и т.д хранятся на главной диагонали* miss[i][i] = miss[i][n];  
 }  
 **for** (i = 0; i < n; i++) {  
 **for** (j = 0; j < n; j++) {  
 **if** (j != i) {  
 data[i][i] += data[i][j] \* helper[j];  
 miss[i][i] = miss[i][i] + fabs(data[i][j]) \* miss[j][j] + fabs(helper[j]) \* miss[i][j];  
 }  
 }  
   
 }  
 help = 0;  
 **for** (i = 0; i < n; i++) {  
 **if** (fabs(fabs(h2[i]) - fabs(data[i][i])) > E) {  
 help = 1;  
 **break**;  
 }  
 }  
 printf(**"\nВаша расширенная матрица: "**);  
 **for** (i = 0; i < n; i++) {  
 printf(**"\n"**);  
 **for** (j = 0; j < n + 1; j++)  
 printf(**"%5.5lf "**, data[i][j]);  
 }  
 } **while** (help);  
 printf(**"\n\nКорни уравнения:"**);  
 **for** (i = 0; i < n; i++)  
 *// printf("\nx%d = %lf (погрешность %lf)", i, data[i][i], miss[i][i]);* printf(**"\nx%d = %lf"**, i, data[i][i]);  
 **return** 0;  
}  
**void** change(**double**\*\* nc, **int** i1, **int** i2, **int** n)  
{  
 **int** j;  
 **double** buf;  
 **for** (j = 0; j < n + 1; j++) {  
 buf = nc[i1][j];  
 nc[i1][j] = nc[i2][j];  
 nc[i2][j] = buf;  
 }  
}

Скриншоты программы:



а) Решение исходной задачи

б) Решение альтернативных исходных данных