

```
1  #include <iostream>
2  #include <math.h>
3  #include <sys/time.h>
4  #include <stdio.h>
5
6  using namespace std;
7  int N; // rozmiar wczytanej macierzy
8  long double m[101][101]; // wczytana macierz
9  long double solution[101]; // wzorcowa odpowiedź
10 long double wyniki[101];
11 long double X[101];
12 using namespace std;
13
14 /* Znajdź wiersz z największą wartością w kolumnie start.
15  * Zamień go miejscem z wierszem nr start.
16  */
17 void ZamienWiersz(int a, int b) {
18     long double* temp = new long double[N + 1];
19     for (int i = 0; i < N + 1; i++) {
20         temp[i] = m[a][i];
21         m[a][i] = m[b][i];
22         m[b][i] = temp[i];
23     }
24     delete[] temp;
25 }
26
27 void ZamienKolumne(int a, int b) {
28     long double* temp = new long double[N + 1];
29     for (int i = 0; i < N + 1; i++) {
30         temp[i] = m[i][a];
31         m[i][a] = m[i][b];
32         m[i][b] = temp[i];
33     }
34     delete temp;
35 }
36
37 void WyborCzesciowy(int start) {
38     int max_pos = start;
39     long double max_value = -100000;
40     for (int i = start; i < N; i++) {
41         if (max_value < m[i][start]) {
42             max_value = m[i][start];
43             max_pos = i;
44         }
45     }
46
47     //cout << "I: " << max_value << endl;
48     if (max_pos == start)
49         return;
50     else ZamienWiersz(max_pos, start);
51 }
52
53
54 void WyborPelny(int start) {
55     int max_posx = start;
56     int max_posy = start;
57     long double max_value = -100000;
58     for (int i = start; i < N; i++) {
59         for (int q = start; q < N; q++) {
60             if (max_value < m[i][q]) {
61                 max_value = m[i][q];
62                 max_posx = i;
```

```
63         max_posy = q;
64     }
65 }
66
67 }
68 if (max_posx == start && max_posy == start)
69     return;
70 else {
71     ZamienWiersz(max_posx, start);
72     ZamienKolumne(max_posy, start);
73 }
74 }
75
76 /* Odejmij wielokrotności wiersza [start] od reszty macierzy tak,
77 * aby w kolumnie [start] pojawiły się 0.
78 */
79 void ZerujKolumne(int start) {
80     for (int i = start + 1; i < N; i++) {
81         long double wiel = m[i][start] / m[start][start];
82         for (int ii = start; ii < N + 1; ii++) {
83             if (m[start][ii] != 0) {
84                 m[i][ii] = m[i][ii] - wiel * m[start][ii];
85             }
86         }
87     }
88 }
89 }
90
91 void PokazMacierz() {
92     for (int i = 0; i < N; i++) {
93         for (int q = 0; q < N + 1; q++)
94             cout << m[i][q] << " ";
95         cout << endl;
96     }
97     cout << endl;
98 }
99
100
101 /* Dzieli cały wiersz przez współczynnik znajdujący się na przekątnej i otr-
102 * zymujemy same jedynki na przekątnych.
103 * Odejmujemy od wyniku ( ostatniej kolumny ) wartości współczynników wymno-
104 * żone przez znaną odpowiedź z wektora X.
105 */
106 void Rozwiaz() {
107     for (int i = N - 1; i >= 0; i--) {
108         long double temp = m[i][i];
109         for (int q = i; q <= N; q++) {
110             m[i][q] = m[i][q] / temp;
111         }
112         long double wynik = m[i][N];
113         for (int q = i + 1; q < N; q++) {
114             wynik -= X[q] * m[i][q];
115         }
116         X[i] = wynik;
117     }
118 }
119
120
121 int main() {
122
```

```
123     struct timeval start, end;
124     long mtime, seconds, useconds;
125
126     int ilosc;
127     cin >> ilosc;
128
129     int old = 0;
130     gettimeofday(&start, NULL);
131
132     while (ilosc--) {
133         cin >> N;
134         // Zerujemy macierz
135         for (int i = 0; i < 101; i++) {
136             for (int q = 0; q < 101; q++)
137                 m[i][q] = 0;
138             solution[i] = 0;
139             X[i] = 1;
140         }
141
142         // Obsługujemy mierzenie czasu
143
144         if (old == 0) old = N;
145         if (old != N) {
146             gettimeofday(&end, NULL);
147             seconds = end.tv_sec - start.tv_sec;
148             useconds = end.tv_usec - start.tv_usec;
149
150             mtime = ((seconds) * 1000 + useconds / 1000.0) + 0.5;
151
152             printf("%ld\n", mtime);
153             old = N;
154             gettimeofday(&start, NULL);
155         }
156         // Wprowadzamy dane
157         for (int i = 0; i < N; i++) {
158             cin >> solution[i];
159         }
160         int nic;
161         cin >> nic;
162         for (int i = 0; i < N; i++)
163             for (int q = 0; q < N + 1; q++)
164                 cin >> m[i][q];
165
166         // Rozpoczynamy liczenie
167         for (int i = 0; i < N; i++) {
168             //PokazMacierz();
169             //WyborCzesciowy(i);
170             WyborPelny(i);
171             ZerujKolumne(i);
172         }
173         Rozwiaz();
174         for (int i = 0; i < N; i++)
175             for (int i = 0; i < N; i++)
176                 wyniki[N] += solution[i] - X[i];
177     }
178
179     for (int i = 0; i < N; i++)
180         printf("%.20Lf\n", (long double) fabs(wyniki[i]));
181
182     return 0;
183 }
184
```