```cpp
//Metoda gradientow sprzezonych
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

int main() {
  int dim = 128;
int iteracji = 40;
  double A[dim][dim] = {0};
  double x[dim];
  double e[dim];
  double p[dim];
  double r[dim];
  double Ap[dim];
  double rr[dim];
  double pp[dim];
  double norma[dim];
  double nnorma[dim];

  for(int i=0; i<dim; i++){
      x[i] = e[i] = norma[i] = 1;
  for(int j=0; j<dim; j++) {
    if(i==j) {
     A[i][j] = 4;
     if(j<dim-1) A[i+1][j] = A[i][j+1] = 1;
     if(j<dim-4) A[i+4][j] = A[i][j+4] = 1;
      } } }

  r[0] = e[0]+(-A[0][0]*x[0]-A[0][1]*x[1]-A[0][4]*x[4]);
  r[1] = e[1]+(-A[1][0]*x[0]-A[1][1]*x[1]-A[1][2]*x[2]-A[1][5]*x[5]);
  r[2] = e[2]+(-A[2][1]*x[1]-A[2][2]*x[2]-A[2][3]*x[3]-A[2][6]*x[6]);
  r[3] = e[3]+(-A[3][2]*x[2]-A[3][3]*x[3]-A[3][4]*x[4]-A[3][7]*x[7]);

  for(int j=4; j<dim-4; j++) r[j] = e[j]-(A[j][j-4]*x[j-4]+A[j][j-1]*x[j-1]+A[j][j]*x[j]+A[j][j+1]*x[j+1]+A[j][j+4]*x[j+4]);

  r[dim-4] = e[dim-4]+(-A[dim-4][dim-8]*x[dim-8]-A[dim-4][dim-5]*x[dim-5]-A[dim-4][dim-4]*x[dim-4]-A[dim-4][dim-3]*x[dim-3]);
  r[dim-3] = e[dim-3]+(-A[dim-3][dim-7]*x[dim-7]-A[dim-3][dim-4]*x[dim-4]-A[dim-3][dim-3]*x[dim-3]-A[dim-3][dim-2]*x[dim-2]);
  r[dim-2] = e[dim-2]+(-A[dim-2][dim-6]*x[dim-6]-A[dim-2][dim-3]*x[dim-3]-A[dim-2][dim-2]*x[dim-2]-A[dim-2][dim-1]*x[dim-1]);
  r[dim-1] = e[dim-1]+(-A[dim-1][dim-5]*x[dim-5]-A[dim-1][dim-2]*x[dim-2]-A[dim-1][dim-1]*x[dim-1]);

  for(int i=0; i<dim; i++)
    p[i] = r[i];

  for(int i=0; i<iteracje; i++) {
      Ap[0] = A[0][0]*p[0]+A[0][1]*p[1]+A[0][4]*p[4];
    Ap[1] = A[1][0]*p[0]+A[1][1]*p[1]+A[1][2]*p[2]+A[1][5]*p[5];
    Ap[2] = A[2][1]*p[1]+A[2][2]*p[2]+A[2][3]*p[3]+A[2][6]*p[6];
    Ap[3] = A[3][2]*p[2]+A[3][3]*p[3]+A[3][4]*p[4]+A[3][7]*p[7];

     for(int j=4; j<dim-4; j++) Ap[j] = A[j][j-4]*p[j-4]+A[j][j-1]*p[j-1]+A[j][j]*p[j]+A[j][j+1]*p[j+1]+A[j][j+4]*p[j+4];

     Ap[dim-4] = A[dim-4][dim-8]*p[dim-8]+A[dim-4][dim-5]*p[dim-5]+A[dim-4][dim-4]*p[dim-4]+A[dim-4][dim-3]*p[dim-3];
     Ap[dim-3] = A[dim-3][dim-7]*p[dim-7]+A[dim-3][dim-4]*p[dim-4]+A[dim-3][dim-3]*p[dim-3]+A[dim-3][dim-2]*p[dim-2];
```

```
55      Ap[dim-2] = A[dim-2][dim-6]*p[dim-6]+A[dim-2][dim-3]*p[dim-3]+A[dim-
    2][dim-2]*p[dim-2]+A[dim-2][dim-1]*p[dim-1];
56      Ap[dim-1] = A[dim-1][dim-5]*p[dim-5]+A[dim-1][dim-2]*p[dim-2]+A[dim-
    1][dim-1]*p[dim-1];
57
58      double l, m, a, b;
59      l = m = a = b = 0;
60
61      for(int j=0; j<dim; j++) {
62        l = l + r[j]*r[j];
63         m = m + p[j]*Ap[j];  }
64      a = l/m;
65      for(int j=0; j<dim; j++) {
66        rr[j] = r[j] - a*Ap[j];
67        x[j] = x[j] + a*p[j]; }
68    m = l;
69    l = 0;
70    for(int j=0; j<dim; j++)
71        l = l + rr[j]*rr[j];
72    b = l/m;
73    for(int j=0; j<dim; j++)
74        pp[j] = rr[j] + b*p[j];
75    for(int j=0; j<dim; j++) {
76        p[j] = pp[j];
77        r[j] = rr[j]; }
78    for(int j=0; j<dim; j++)
79        nnorma[j] = x[j];
80    double normaa = 0;
81    for(int j=0; j<dim; j++)
82    normaa = normaa+(nnorma[j]-norma[j])*(nnorma[j]-norma[j]);
83    normaa = sqrt(normaa);
84    for(int j=0; j<dim; j++) norma[j] = nnorma[j];
85    if(i>=1) cout << setprecision(12) << fixed << "||xk - x(k-1)|| = " <<
    normaa << endl;}
86     for(int i=0; i<dim; i++) cout << "x" << i << " = " << x[i] << endl;
87    return 0;}
88
```