Zadanie 1 – rozwiązanie układu równań z uwzględnieniem struktury macierzy.
Do rozwiązania tego zadania napisałam program w języku C++ z wykorzystaniem biblioteki GSL (Gnu Scientific Library).
a)

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

Jako iż macierz jest trójdiagonalna, w celu optymalnego przechowania jej w pamięci, utworzyłam dwa wektory:
d - do przechowania głównej diagonali
e – do przechowania naddiagonali i poddiagonali, które są identyczne
Ponadto wektor b przechowuje prawe strony równań, natomiast wektor x po wykonaniu programu będzie zawierał wartości dla kolejnych x1, x2...x7.
Następnie wywołuję funkcję gsl_linalg_solve_symm_tridiag(d, e, b, x), która według dokumentacji GSL, rozwiązuje układ równań postaci Ax=b dla macierzy A symetrycznej, trójdiagonalnej i dodatnio określonej. Funcja ta wykorzysuje faktoryzację Cholesky'ego, uwzględniając strukturę macierzy.

Uzyskane wyniki to:
x1 = 0.166789
x2 = 0.332842
x3 = 0.501841
x4 = 0.659794
x5 = 0.858984
x6 = 0.904271
x7 = 1.52393

b)

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

Tutaj macierz również jest symetryczna, trójdiagonalna i dodatnio określona, a ponadto jest macierzą cykliczną, która w elementach $a_{1,n}$ i $a_{n,1}$ przechowuje ostatni element wektora e, który tym razem musi przechowywać n elementów (zamiast n-1 jak to było w podpunkcie a).

Zgodnie z dokumentacją GSL, dla macierzy takiej postaci (jak na zdjęciu obok, zdjęcie skopiowane z dokumentacji GSL), należy użyć funkcji gsl_linalg_solve_symm_cyc_tridiag(d, b, e, x). Funkcja ta również wykorzystuje algorytm Cholesky'ego, uwzględniając strukturę macierzy.

$$A = \begin{pmatrix} d_0 & e_0 & 0 & e_3 \\ e_0 & d_1 & e_1 & 0 \\ 0 & e_1 & d_2 & e_2 \\ e_3 & 0 & e_2 & d_3 \end{pmatrix}$$

W tym przypadku, uzyskałam wyniki:
x1 = -0.260163
x2 = 0.447154
x3 = 0.471545
x4 = 0.666667
x5 = 0.861789
x6 = 0.886179
x7 = 1.5935