Wyznaczanie wartości i wektorów własnych macierzy symetrycznej.

Zadania do wykonania:

1. Zdefiniować macierz symetryczną A o wymiarze $n = 5$, której elementy są dane przepisem:

$$A_{ij} = \sqrt{(i+1)+(j+1)} \tag{1}$$

   gdzie: $i, j = 0, 1, 2, 3, 4$.

2. Dokonać redukcji macierzy do postaci trójdiagonalnej ($A \to T$) przy użyciu procedury:

```
dsytrd_(char *uplo, int *n, double *a, int *lda, double *d__, double *e,
double *tau, double *work, int *lwork, int *info);
```

   z biblioteki LAPACK dla C.
   a - to macierz którą diagonalizujemy (A), d__ i e to wektory odpowiednio n i (n-1)-elementowe w których zapisane są składowe diagonali i poddiagonali macierzy wynikowej.
   W pliku 'dodatki.cpp' znajduje się wrapper dla funkcji dsytrd_, który upraszcza przekazywanie argumentów.
   Macierz A przekształciliśmy do postaci iloczynu:

$$T = P^{-1}AP \tag{2}$$

3. Zapisać do pliku tekstowego macierz przekształcenia P.
   Uwaga: dsytrd_ nie zwraca macierzy P. Do jej otrzymania potrzebne jest wywołanie procedury

```
dorgtr_(char *uplo, int *n, double *a, int *lda, double *tau,
double *work, int *lwork, int *info);
```

   która przekształca macierz a otrzymaną z dsytrd_ na macierz P. Funkcja dsytrd_wrapper w 'dodatki.cpp' wywołuje zarówno dsytrd_ jak i dorgtr_ więc ostatecznie nadpisuje macierz a macierzą P.

4. Przy użyciu procedury

```
dsteqr_(char *compz, int *n, double *d__, double *e,
double *z__, int *ldz, double *work, int *info);
```

   (wrapper w dodatki.cpp) znaleźć wartośći i wektory własne macierzy trójdiagonalnej T.

$$T \cdot y = \lambda y \tag{3}$$

5. Zapisać do pliku tekstowego wartości własne macierzy T.

6. Zapisać do pliku tekstowego wektory własne macierzy T.

7. Chcemy znaleźć wektory własne macierzy A więc musimy przekształcić wektory własne macierzy T (dlaczego?)

$$T = P^{-1}AP \tag{4}$$

$$Ty = \lambda y \tag{5}$$

$$x = Py \tag{6}$$

$$Ax = \lambda x \tag{7}$$

   Wektory własne macierzy A zapisać do pliku.

8. Sprawdzić czy rzeczywiście wektory x są wektorami własnymi macierzy A tzn. należy policzyć:

$$\beta_k = \frac{(x_k, Ax_k)}{(x_k, x_k)} \tag{8}$$

   gdzie: (x, Ax) jest iloczynem skalarnym A macierzowym, a (x, x) jest iloczynem skalarnym dwóch wektorów w przestrzeni euklidesowej.

9. Zapisać do pliku tekstowego wartości β_k. W sprawozdaniu porównać z wartościami λ_k. Ewentualne różnice skomentować.

Punktacja: podpunkty (1-3) 40pkt., (4-6) 30pkt., (7) 15pkt., (8-9) 15pkt.

Wskazówki:

- na potrzeby LAPACKa wracamy do indeksowania macierzy od 0 (w odróżnieniu od Numerical Recipes)
- dokumentacja użytych w zadaniu funkcji znajduje się tu: www.netlib.org/clapack/old/double
- każdą z użytych procedur należy zadeklarować

```
extern "C" int dsytrd_(char *uplo, int *n,  .....
```

  itd. Wszystkie potrzebne deklaracje znajdują się w pliku 'dodatki.cpp'.
- przy kompilacji dodajemy linkowanie '-llapack -lblas' (patrz makefile)
- LAPACK wymaga macierzy dwuwymiarowych A[n][n] zapisanych jako macierz jednowymiarowa A[n*n] (column-wise). Funkcje przepisujące macierz z 2D na 1D i na odwrót znajdują się w 'dodatki.cpp'.
- w 'dodatki.cpp' znajdują się także wrappery procedur LAPACKowych, tzn. funkcje upraszczające korzystanie z tych procedur