Damian Szopiński 185394 **Maciej Pestka 170088**

Metoda eliminacji Gaussa:

Uwaga, program wymaga stdc++11, albo nowszej.
Plik “gauss.h” zawiera funkcję GaussElimination zwracającą rozwiązanie macierzy odczytanej z pliku o ścieżce podanej w argumencie, rozwiązaną metodą eliminacji Gaussa.

Na początku zczytywane są wszystkie dane z pliku do jednej tablicy dynamicznej jednowymiarowej.
Na podstawie liczby cyfr odczytanych przez program, łatwo można odczytać wielkość macierzy.
Po obliczeniu N jesteśmy w stanie przetłumaczyć macierz jednowymiarową na macierz dwuwymiarową.
Następnie wykonujemy prodedurę przestawiania wierszy. Jesteśmy w stanie wyłączyć tę część kodu ustawiając drugi argument przestawianieWierszy=false przy wywoływaniu funkcji. Będzie to przydatne dla pokazania kolejnych punktów zadania.
Następnie przemieniamy macierz, w taki sposób by w następnym kroku rozwiązania równania były łatwe do odczytania.
Na końcu funkcja zwraca wynik w postaci vector<float>.

```
    std::vector<float> arr=GaussElimination("arr.txt",false);
    for (float a:arr)
        std::cout<<a<<'\t';
```

W wyniku wywołania tego fragmentu kodu, funkcja spróbuje zwrócić rozwiązania macierzy z pliku “arr.txt”, bez przestawiania wierszy (argument false)
Uwaga, macierze A i B są połączone na zasadzie

A(0,0)	A(0,1)	A(0,2)	B(0)
A(1,0)	A(1,1)	A(1,2)	B(1)
A(2,1)	A(2,1)	A(2,2)	B(2)

(Kolejne kolumny rozdzielone są tabulacją, a wiersze enterem)
Następnie sprawdziliśmy wyniki dla poniższych macierzy
matrix1.txt

0	1	1	1
1	1	1	2
1	1	0	0

W powyższym przypadku, natrafiamy na 0 na przekątnej. W konsekwencji, podczas wykonywania programu bez przestawiania wierszy program będzie wykonywał operacje dzielenia przez zero. W tym przypadku program zwróci wartości NaN.
W przypadku przestawiania wierszy, program jest w stanie poprawnie wyliczyć rozwiązanie tych równań.

matrix2.txt

9999999999999999 99999999999999	0.000000000000000 0000000000001	0.000000000000000 0000000000001	0.000000000000000 001
1	9999999999999999 99999999999999	0.000000000000000 0000000000001	0.000000000000000 002
1	1	9999999999999999 99999999999999	0.000000000000000 003

Program jest w stanie poprawnie wczytać wartości tych liczb oraz wyświetlić je na ekranie, ale w wyniku wykonywanych operacji (dzielenia małych liczb przez duże liczby, a także mnożenia przez małe liczby podczas wyznaczania wyników) otrzymujemy same zera. Jest to spowodowane tym, że float traci swoją dokładność w tak małych liczbach.

przykład.txt

2	2	6	4
2	1	7	6
-2	-6	-7	-1

Obliczając ręcznie wychodzi nam wynik x1=0, x2=-1, x3=1
Program jest w stanie obliczyć poprawnie wynik w przypadku przestawiania wierszy, jak i bez przestawiania wierszy.

Macierze "pw1.txt" "pw2.txt" "pw3.txt" "pw4.txt" "pw5.txt" "pw6,txt"
To te same macierze, z wszystkimi możliwtymi przestawieniami wierszy.
Program pokazuje takie same rozwiązania co oznacza, że algorytm przestawiania wierszy został napisany poprawnie.

```
Eliminacja Gaussa:
Przypadki w ktorych eliminacja zawodzi:
Macierz 1:
nan     nan     nan


Macierz 2:
0       0       0


Przyklad poprawny:
0       -1      1


Przyklad z wykladu:
1.625   -0.375  0.25


Wszystkie mozliwe przestawienia wierszy (macierz pw 3x3)
0       -1      1
0       -1      1
0       -1      1
0       -1      1
0       -1      1
0       -1      1
```

Metoda rozkładu LU:

Na początku program deklaluje tablicę dwójwymiarowe (3,3) oraz jednowymiarowe. W tablicy A będzie zapisywania macierz, a w macierzach L i U rozkład macierzy A na górnotrójkątną i dolną trójkątną.
Następnie program wylicza macierz L i zapisuję wyniki do tablicy l, tak samo postępuję do obliczania macierzy U. Potem wyświetla macierz A oraz macierze L i U.
Na końcu program wylicza wartości x i y na podstawie macierzy L, U i tablicy B. Na końcu wyświetla na ekranie wyniki końcowe.

Rozkład LU został przetestowany dla macierzy 3x3 który jest poniżej pokazany, podobnie jak dla eliminacji Gaussa.

przykład.txt

2	2	6	4
2	1	7	6
-2	-6	-7	-1

Wyniki wychodzą w programie: (0,-1,1), które są poprawnie.