Instytut Informatyki Politechniki Śląskiej

Zespół Mikroinformatyki i Teorii Automatów Cyfrowych

Rok akademicki:	Rodzaj studiów*: SSI/NSI/NSM	Przedmiot (Języki Asemblerowe/SMiW):	Grupa	Sekcja
2016/2017	**SSI**	**Języki Asemblerowe**	**9**	**-**

Imię:	**Zbigniew**	**Prowadzący**: OA/JP/KT/GD/BSz/GB	**AO**
Nazwisko:	**Kmonk**		

Raport końcowy

Temat projektu:

Wyznaczanie rzędu macierzy

Data oddania: dd/mm/rrrr	**25/02/2017**

1. Temat projektu i opis założeń z rozbiciem na założenia części głównej programu oraz funkcje biblioteki.

Tematem projektu jest wyznaczanie rzędu macierzy kwadratowej o dowolnym rozmiarze. Wyznaczniki macierzy, z których korzystamy podczas wyznaczania rzędu obliczane są za pomocą rozwinięcia Laplace'a. Odpowiadają za to dwie biblioteki DLL (jedna napisana w języku C++, druga w ASM). Program główny, napisany w języku C++, odpowiada za część wejście/wyjście programu. Wczytuje ona z pliku macierze, otwiera biblioteki, wyświetla wyniki, przydziela ilość wątków oraz wyświetla wyniki. Program uruchamiany jest z linii poleceń i nie posiada interfejsu graficznego, wszystkie informacje wyświetlane są w linii poleceń. Zarówno program jak i biblioteki występują jedynie w wersji x64.

2. Analiza zadania w tym uzasadnienie wyboru rozwiązania.

Celem programu jest wyznaczenie rzędu macierzy. Proces ten polega na odnalezieniu największej podmacierzy, której wyznacznik jest różny od zera. Wielkość takiej niezerowej macierzy będzie szukanym przez nas rzędem. Przykładowo:

$$A = \begin{bmatrix} 2 & 8 & 3 & -4 \\ 1 & 4 & 1 & -2 \\ 5 & 20 & 0 & -10 \\ -3 & -12 & -2 & 6 \end{bmatrix}$$

Wyznacznik tej macierzy jest równy zero, więc musimy wyznaczyć podmacierze, poprzez skreślenie jednej z kolumn i jednego z wierszy. Dla macierzy o wielkości *n*, można w ten sposób wyznaczyć *n* do kwadratu podmacierzy, czyli w wypadku tej macierzy, szesnaście.

$$\begin{bmatrix} 2 & 8 & 3 & -4 \\ 1 & 4 & 1 & -2 \\ 5 & 20 & 0 & -10 \\ -3 & -12 & -2 & 6 \end{bmatrix} \text{ czyli } \begin{bmatrix} 4 & 1 & -2 \\ 20 & 0 & -10 \\ -12 & -2 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 8 & 3 & -4 \\ 1 & 4 & 1 & -2 \\ 5 & 20 & 0 & -10 \\ -3 & -12 & -2 & 6 \end{bmatrix} \text{ czyli } \begin{bmatrix} 1 & 1 & -2 \\ 5 & 0 & -10 \\ -3 & -2 & 6 \end{bmatrix} \text{ itd.}$$

W tym wypadku, wyznaczniki wszystkich szesnastu macierzy są równe zero, więc musimy kontynuować wyznaczanie kolejnych podmacierzy, przez skreślenie kolejnych wierszy i kolumn w każdym przypadku.

$$\begin{bmatrix} 2 & 8 & 3 & -4 \\ 1 & 4 & 1 & -2 \\ 5 & 20 & 0 & -10 \\ -3 & -12 & -2 & 6 \end{bmatrix} \text{ czyli } \begin{bmatrix} 0 & -10 \\ -2 & 6 \end{bmatrix}$$

$$det(A) = \begin{vmatrix} 0 & -10 \\ -2 & 6 \end{vmatrix} = 0 \cdot 6 - (-10) \cdot (-2) = 0 - 20 = -20$$

Tym razem uzyskaliśmy macierz, której wyznacznik jest różny od zera, tak więc możemy stwierdzić, iż rząd naszej macierzy jest równy dwa.

Obliczanie wyznaczników macierzy większych niż trzy na trzy jest jednak dość kłopotliwe bez odpowiednich algorytmów, dlatego też korzystamy z rozwinięcia Laplace'a.

"Dla każdej macierzy A o wymiarach n×n wyznacznik detA spełnia regułę zwaną rozwinięciem Laplace'a. Wyznacznik macierzy A jest równy sumie iloczynów elementów dowolnie wybranego wiersza (kolumny) przez ich dopełnienia algebraiczne. Rozwinięcie Laplace'a pozwala obliczyć wyznacznik macierzy unikając korzystania z bardzo czasochłonnej metody opartej na definicji wyznacznika."

W postaci ogólnej możemy zapisać to następująco:

$$\det A = \sum_{j=1}^{n} a_{ij} A ij$$

gdzie:

j - to numer kolumny względem którego rozwijamy dany wyznacznik
a_{ij} - element macierzy w ***i***-tym wierszu i ***j***-tej kolumnie
A_{ij} - to dopełnienie algebraiczne elementu a_{ij} (powstałe z przemnożenia czynnika $(-1)^{i+j}$ przez minor elementu a_{ij})

Przykładowo:

$$W = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$W = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13} =$$

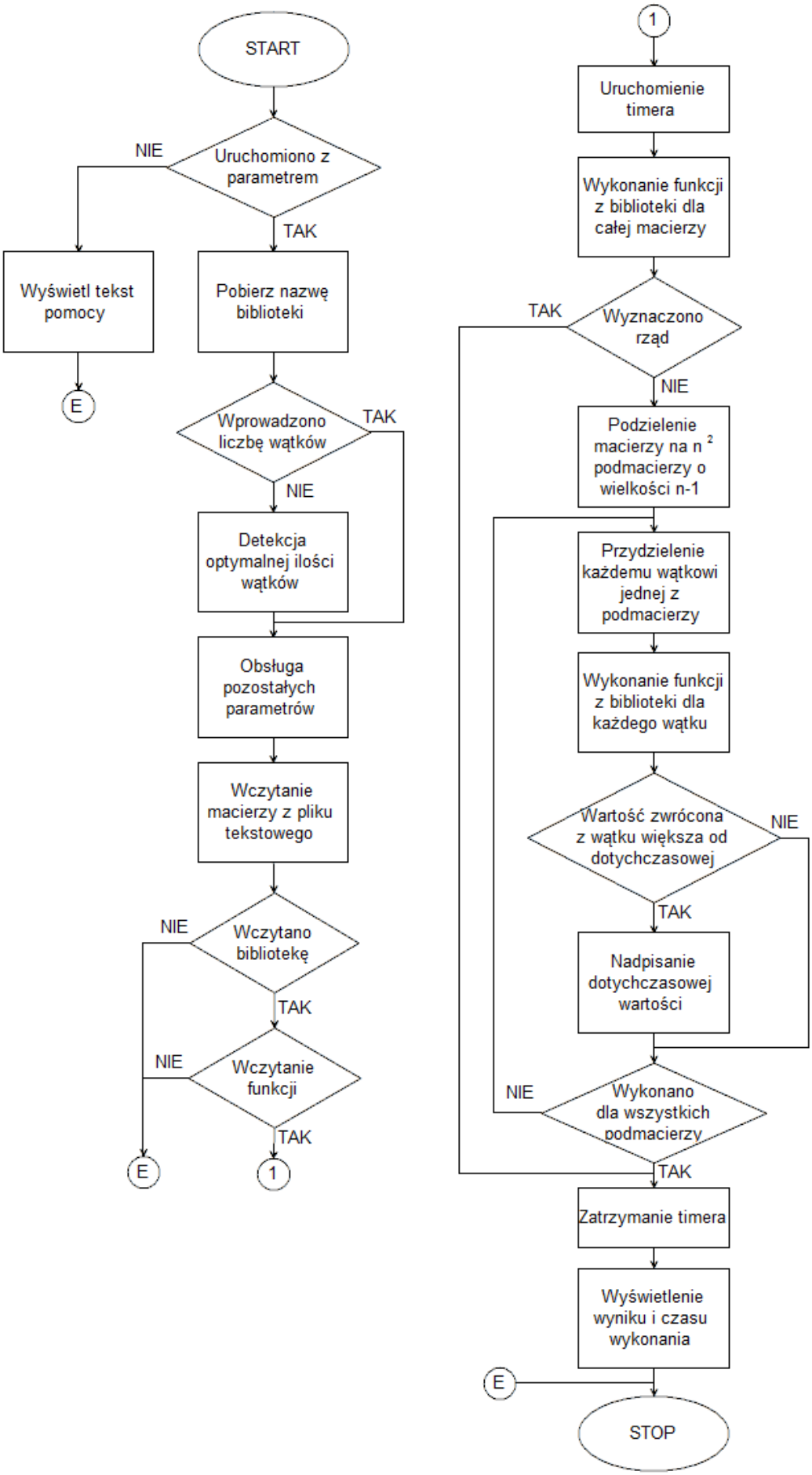
$$= (-1)^{1+1} a_{11} M_{11} + (-1)^{1+2} a_{12} M_{12} + (-1)^{1+3} a_{13} M_{13} =$$

$$= a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} =$$

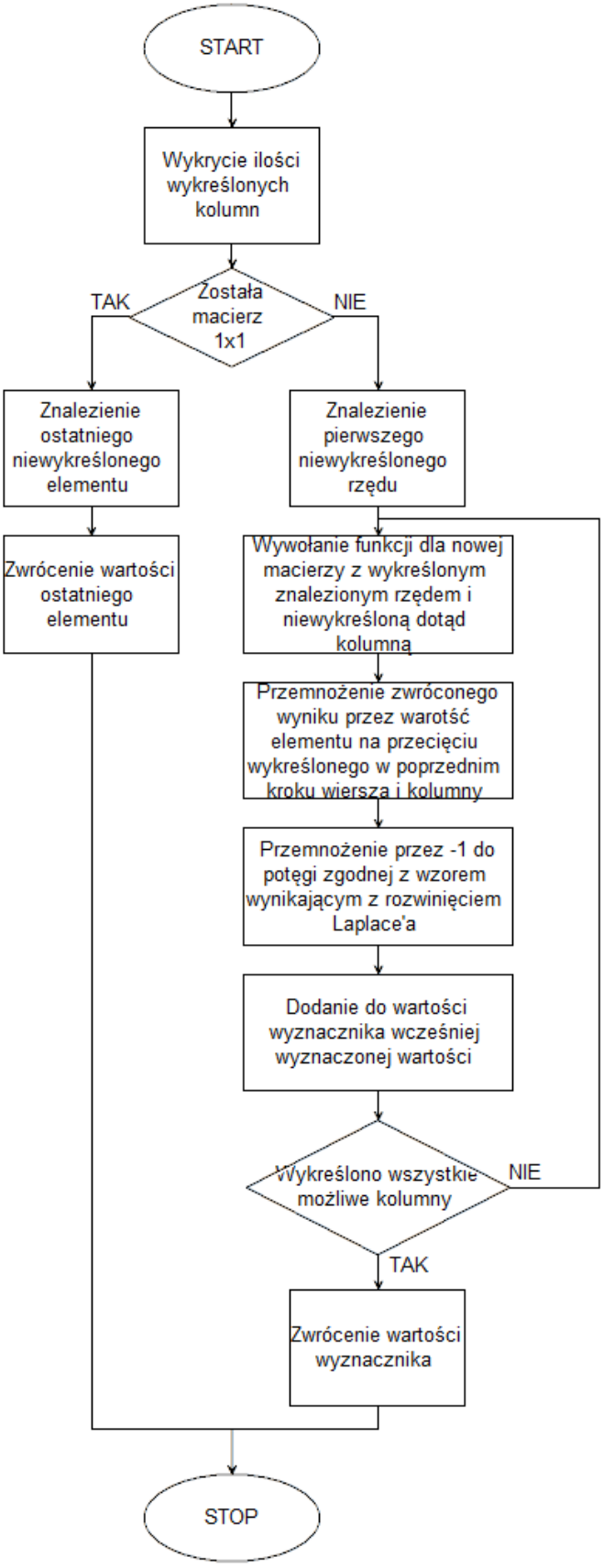
$$= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

3. Schemat blokowy programu.

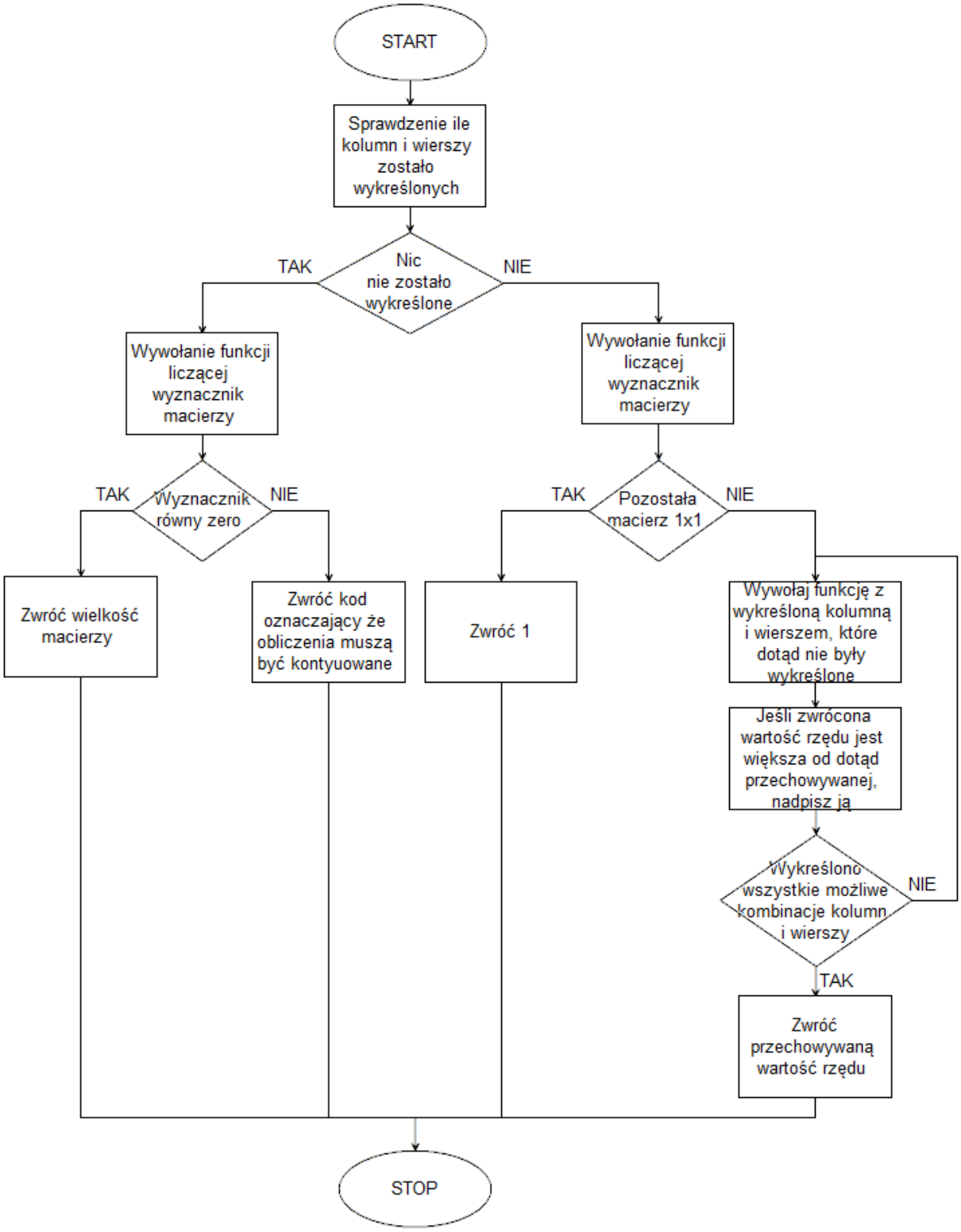
Schemat głównego programu:



Schemat działania funkcji liczenia wyznacznika z bibiotek (pomiędzy wersjami ASM a C++ nie ma różnic w logice rozwiązania):



Schemat działania funkcji liczącej rząd z bibliotek (pomiędzy wersjami ASM a C++ nie ma różnic w logice rozwiązania):



4. Opis programu w języku wysokiego poziomu i zmiennych globalnych.

Program główny zawiera następujące funkcje:

- `void __cdecl ThreadProc(void * Args)` - funkcja pomagająca przekazywać kilka argumentów do wątków. Jeśli wątek zwróci większą wartość niż inne wątki, nadpisuje zmienną globalną będącą wynikiem.
- `int _tmain(int argc, _TCHAR* argv[])` - główna funkcja, obsługująca wczytywanie danych, parametry, tworzenie wątków wyświetlanie wyniku oraz testowanie bibliotek.

Zmienne globalne:

- `MYPROC dllFunction` - zawiera adres funkcji z załadowanej biblioteki DLL
- `int value` - wartość rzędu macierzy, nadpisywana przez wątki, gdy zwracają większą wartość

Struktury:

- `typedef int(*MYPROC)(int *, int, int)` - struktura potrzebna by określić funkcję wczytywaną z biblioteki
-
```
typedef struct {
	int * matrix;
	int deleteTemplateColumn, deleteTemplateRow;
	int rank;
}Testing;
```
- struktura do przekazywania wątkom kilku argumentów

5. Opis funkcji bibliotek DLL w językach C++ oraz ASM i ich parametrów oraz używanych rejestrów.

C++:

- `int rank(int * matrix, int deleteTemplateColumn, int deleteTemplateRow)`

Funkcja przyjmuje jako parametry adres do macierzy, maskę wykreślonych kolumn i maskę wykreślonych wierszy. Działanie funkcji zostało przedstawione w punkcie 4. Funkcja ma dwa tryby działania. Dla macierzy bez wykreślonych rzędów i kolumn, wykonuje się tylko raz. Po otrzymaniu z funkcji `determinant` wyznacznika macierzy, zwraca albo wielkość macierzy, albo -2 oznaczające, że wyznacznik macierzy jest równy zero. Kiedy w macierzy są wykreślone rzędy i kolumny, przechodzi do rekurencyjnego trybu wyznaczania rzędu macierzy.

- `int determinant(int * matrix, int deleteTemplateColumn, int deleteTemplateRow)`

Funkcja przyjmuje jako parametry adres do macierzy, maskę wykreślonych kolumn i maskę wykreślonych wierszy. Działanie funkcji zostało przedstawione w punkcie 4. W funkcji wykorzystywane jest rozwinięcie Laplace'a w celu obliczenia wyznacznika macierzy. Funkcja zwraca obliczony rekurencyjnie wyznacznik macierzy.

ASM:

- `rank proc`

Funkcja działa identycznie jak w wypadku odpowiadającej jej funkcji w C++. Wykorzystuje rejestr MM7 w celu przechowywania wartości potrzebnej do wektorowych operacji logicznych, oraz rejestry MM0-2 i MM6 do obliczeń i innych operacji wektorowych. Rejestr R10 jest wykorzystywany do przechowywania wartości ilości wykreślonych kolumn. Pozostałe rejestry przechowują wartości tymczasowe.

- `determinant proc`

Funkcja działa identycznie jak w wypadku odpowiadającej jej funkcji w C++. Wykorzystuje rejestr MM7 w celu przechowywania wartości potrzebnej do wektorowych operacji logicznych, oraz rejestry MM0-1 i MM6 do obliczeń i innych operacji wektorowych. Ważne są też rejestry R10, R9, RCX oraz R11, przechowujące wartości potrzebne do zachowania kolejności obliczeń w pętlach po powrocie z obliczeń rekurencyjnych.

6. Opis struktury danych wejściowych/testowych programu.

Dane wejściowe muszą być zawarte w folderze "Dane testowe". Jeśli za pomocą parametrów nie zostanie podana inna nazwa pliku, otworzony zostanie plik "example.txt". Dane wejściowe powinny być w następującym formacie:

6;7;8;
0;0;0;
3;3;3;

lub

6;7;8;0;0;0;3;3;3;

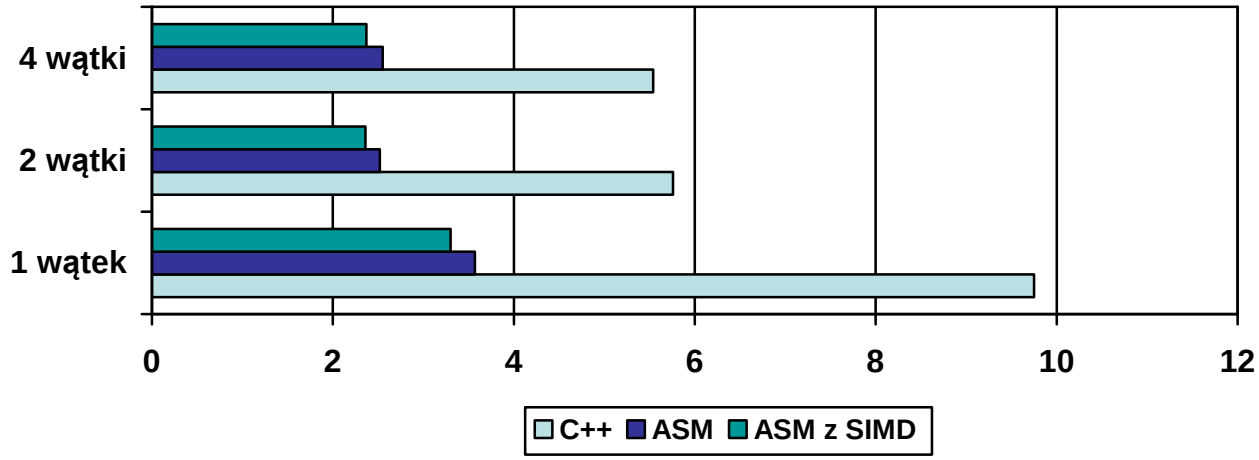
Należy zapewnić, aby pierwiastek z liczby wpisanych elementów był liczbą naturalną.

7. Opis uruchamiania programu i jego testowania.

Program uruchamiany jest z linii poleceń i może przyjmować następujące parametry:
- pierwszym parametrem musi być nazwa biblioteki
- jeśli chcemy określić liczbę wątków, należy podać ilość wątków w formacie liczbowym jako drugi parametr
- parametr "-f filename" pozwala na wybranie innego pliku z danymi wejściowymi, przy czym filename zastępujemy nazwą naszego pliku wejściowego
- parametr "-d" pozwala nam na wejście w tryb debugu, gdzie wyświetlane jest więcej informacji
- parametr "-t" pozwala na wejście w tryb testów, gdzie rząd macierzy obliczany jest sto razy

Program obsługuje błędy wywołane z powodu błędnego wprowadzenia parametrów, braku plików bibliotek, braku potrzebnej funkcji oraz błędnego wypełnienia pliku wejściowego.

8. Wyniki pomiarów czasu wykonania programu dla różnych danych testowych i obu wersji bibliotek przedstawione na wykresach porównawczych.



Na wykresie przedstawiony jest średni czas wykonywania dla macierzy 7x7. Dla większych macierzy, czasy są zbyt długie by przeprowadzić ilość testów potrzebnych na wyciągnięcie wniosków, zaś dla macierzy mniejszych zajmuje to zbyt mało czasu.

	C++		ASM		ASM z SIMD	
	MIN	MAX	MIN	MAX	MIN	MAX
1 wątek	9,68	10,31	3,54	3,74	3,28	3,34
2 wątki	5,31	9,94	2,46	3,33	2,28	2,96
4 wątki	5,29	6,84	2,45	3,28	2,29	2,80

Czas zaznaczony na czerwono wyraźnie nie pasuje do reszty czasów. Wywołane jest to faktem, że na komputerze na którym dokonywano testów, obciążenie procesora wynosiło 100%, a w tle otworzył się niechciany program.

9. Analiza działania programu z wykorzystaniem modułu profilera VS2013.

Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
MSVCR120.dll	**1 673**	**1 673**	**65,25**	**65,25**
RzadMacierzyCpp.DLL	**2 564**	**891**	**100,00**	**34,75**
determinant	1 302	533	50,78	20,79
rank	2 564	327	100,00	12,75
pow	31	31	1,21	1,21
RzadMacierzy.exe	**2 564**	**0**	**100,00**	**0,00**
ThreadProc	2 564	0	100,00	0,00

Sampling dla biblioteki C++ z 1 wątkiem dla macierzy 7x7

Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
RzadMacierzyASM.DLL	**809**	**809**	**100,00**	**100,00**
determinant	480	480	59,33	59,33
rank	602	329	74,41	40,67

Sampling dla biblioteki ASM z 1 wątkiem dla macierzy 7x7

Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
MSVCR120.dll	**88 565**	**88 565**	**63,84**	**63,84**
RzadMacierzyCpp.DLL	**138 735**	**50 170**	**100,00**	**36,16**
determinant	69 809	29 701	50,32	21,41
rank	138 735	19 143	100,00	13,80
pow	1 326	1 326	0,96	0,96
RzadMacierzy.exe	**138 735**	**0**	**100,00**	**0,00**
__tmainCRTStartup	1	0	0,00	0,00
ThreadProc	138 734	0	100,00	0,00
wmain	1	0	0,00	0,00

Sampling dla biblioteki C++ z 1 wątkiem dla macierzy 8x8

Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
RzadMacierzyASM.DLL	**60 013**	**60 013**	**100,00**	**100,00**
determinant	36 507	36 507	60,83	60,83
rank	43 699	23 506	72,82	39,17

Sampling dla biblioteki ASM z 1 wątkiem dla macierzy 8x8

Jak widać na zamieszczonych zrzutach ekranu, najwięcej czasu podczas wykonywania programu z biblioteką DLL wykonaną w języku C++ zajmuje MSVCR120.dll, znacząco spowalniając działanie programu. Różnica między czasem wykonywania dla macierzy 8x8 dla obu bibliotek jest naprawdę znacząca. Po około dwunastu minutach biblioteka C++ dalej wykonywała operacje obliczeniowe, podczas gdy biblioteka ASM skończyła działanie po około 5 minutach.

Procentowe wykonanie poszczególnych funkcji nie jest zaskoczeniem - spodziewanym wynikiem było częste wykonywanie funkcji *determinant*. Aby dalej zoptymalizować program, na pewno należałoby skupić się na tej właśnie funkcji.

10. Instrukcja obsługi programu.

Program zawiera w sobie instrukcję obsługi, dostępna z pomocą parametru -h lub -help. Obsługiwane przez aplikację błędy kierują użytkownika w stronę tej pomocy.

Aby poprawnie uruchomić program, należy uruchomić go z poziomu konsoli z parametrem będącym nazwą biblioteki DLL. Drugim parametrem może być liczba wątków, z która mają być uruchamiane obliczenia (w innym wypadku liczba wątków wykrywana jest automatycznie). Parametr -f filename (gdzie filename to nazwa pliku wejściowego), pozwala zmienić plik wejściowy z pliku example.txt, w którym można również zaobserwować poprawny i najwygodniejszy sposób na wprowadzanie danych wejściowych. Dodatkowe, mniej ważne parametry zostały opisane w punkcie 7.

11. Wnioski.

Instrukcje wektorowe nie zawsze przynoszą korzyści w programie, szczególnie gdy możliwość ich użycia jest mocno ograniczona ze względu na rekurencję. Program musi być odpowiednio skonstruowany, by mogły przynieść maksymalne korzyści.
Mimo że tworzenie bibliotek w asemblerze jest dość długim i nużącym ze względu na ograniczone możliwości debuggowania, może przynieść znaczące efekty w czasie wykonywania programu, co doskonale widać w wypadku mojego programu.

12. Literatura.

http://www.naukowiec.org/wiedza/matematyka/rzad-macierzy_611.html
http://www.naukowiec.org/wiedza/matematyka/rozwiniecie-laplace-a_606.html

W raporcie skorzystano z treści zawartych na powyższych stronach w celu objaśnienia zadania wykonywanego przez program.