Temat: Arytmetyka modularna

Drugi raport z postępu prac

Projekt został napisany w języku C++. W drugim etapie dodane zostały do niego nowe algorytmy napisane w języku asembler (Chiński Test Pierwszości, Małe Twierdzenie Fermata).

Pierwszą implementacją w programie jest wyznaczanie NWD za pomocą algorytmu Euklidesa. Aby zobrazować działanie algorytmu, załóżmy że należy wyznaczyć NWD z liczb *a* oraz *b*. Na początku wykonywane jest dzielenie z resztą liczby *a* przez liczbę *b*. Jest to realizowane za pomocą instrukcji *div* w języku assembler. Gdy reszta z dzielenia, która jest umieszczona w rejestrze %edx wynosi 0 to największym wspólnym dzielnikiem jest liczba *b*. W przypadku gdy reszta jest różna od zera to następuje przypisanie liczbie *a* wartości liczby *b*. Następnie liczbie *b* jest przypisywana wartość reszty. Ponownie jest realizowane dzielenie liczby *a* przez *b*, aż reszta nie będzie równa zero. Cała operacja jest realizowana w pętli while, poprzez użycie instrukcji *cmp*. Implementacja algorytmu znajduje się w pliku *Euklides.cpp*. Dodatkowo znajduje się tam również funkcja wyświetlająca menu oraz pobierająca dane od użytkownika.

Drugim algorytmem, który został zaimplementowany jest algorytm bazujący na Małym Twierdzeniu Fermata. Na początku sprawdzana jest podzielność wprowadzonej liczby przez liczby pierwsze z przedziału <2;1000>. Umożliwia to wstępną eliminację liczb złożonych oraz liczb pseudopierwszych Carmichaela. Jeśli liczba będzie złożona, to zwracana jest od razu informacja, że nie spełnia ona zadanego warunku. Następnie w pętli sprawdzany jest warunek Fermata. Zostaje wylosowana podstawa *a*, która zawiera się w przedziale od 2 do testowanej liczby pomniejszonej o jeden. Następnie sprawdzane jest czy *a* jest względnie pierwsza z wprowadzoną liczbą *num*. Polega to inaczej mówiąc na sprawdzeniu, czy *NWD (num, a) = 1*. Jeśli tak, to testowany jest warunek: $a^{num - 1} \bmod num = 1$. Jeśli zostanie spełniony ten warunek, to liczba może być liczbą pierwszą. Jednak nie jest to na tym etapie pewne. Aby to potwierdzić należy sprawdzić to wykonując test Fermata np. dziesięciokrotnie. Liczba, która została uznana liczbą pierwsza może być również liczbą pseudopierwsza Carmichaela. Liczby te są jednak bardzo odległe, a stosując podzielność przez liczby pierwsze z przedziału <2 do 1000> można być pewnym, że algorytm daje poprawne wyniki. Warto zauważyć, że napisane tego algorytmu w języku asembler przysporzyło wielu trudności i wymagało dużej ilości czasu. Co ważne, języki wyższego poziomu pozwalają zapisać warunki w jednej linijce, co w asemblerze jest niemożliwe. Nawet proste porównanie można rozpisać na kilka linijek, co czyni kod bardziej zawiłym i złożonym.

W następnym etapie projektu będziemy chcieli zaimplementować kolejny algorytm.