

Zestaw 3

1. Zadanie treningowe. Napisz klasę, która będzie miała jako składową obiekt `std::string`. Dla tej klasy, podobnie jak to robimy na ćwiczeniach z klasą `TString`, zdefiniuj konstruktor (domyślny, kopiujący, przenoszący), destruktor, operator= (kopiujący, przenoszący), wraz z opcjonalną informacją o działaniu (czyli jak na ćwiczeniach: blok `DEBUG`). Zadeemonstruj w programie różne sytuacje, gdy powyżej napisane komponenty są wywoływane.
2. Napisz klasę o nazwie `BigInt`, która będzie przechowywać dowolnie wielką liczbę (całkowitą) w postaci typu `std::string` oraz pozwalając za pomocą (na razie) metod składowych typu `add(const BigInt&)`, `subtract(const BigInt&)` na obliczenia, zaś metoda `print()` `const` na wypisanie wartości na ekran. Konstruktor oraz operator= oczywiście powinien być obecny. Liczbę podajemy na wejściu jako łańcuch znakowy.
3. Napisz program do mnożenia macierzy o zadanych rozmiarach. Macierz $A_{m \times n}$ oraz $B_{n \times p}$ daje macierz $C_{m \times p}$. Macierze wczytać z pliku ASCII, zapisując w najprostszej możliwej postaci wierszy i kolumn oddzielonych spacjami (np. macierz A będzie mieć m wierszy i n kolumn). Po przemnożeniu wynik zapisać do jakiegoś pliku. Więcej informacji oraz przykładowy program:
https://edufinf.waw.pl/inf/alg/001_search/0074.php
4. Napisz program szyfrujący i deszyfrujący jakąś liczbę za pomocą algorytmu RSA. Polecam przestudiowanie tego materiału: http://michalbereta.pl/do_pobrania/dydaktyka/RSA.pdf a także przykładowej aplikacji: https://edufinf.waw.pl/inf/alg/001_search/0067.php Przy okazji tego zadania można się nauczyć kilku pożytecznych rzeczy oraz napisać parę innych pomocniczych programów (funkcji), np. jak zaproponowano na końcu podanego materiału.
5. Napisz program „kopiujący” bitcoina. Proponuję zacząć od obejrzenia filmu <https://www.youtube.com/watch?v=ZhnJ1bklWWk> (jest tam przykładowy kod w języku Python, ale jest to tak czytelne i proste, że z łatwością da się napisać to samo w języku C++, proszę przyjrzeć się funkcji konwersji https://en.cppreference.com/w/cpp/string/basic_string/to_string). Będziemy potrzebować również SHA265 i tutaj proponuję użyć gotowy i przenośny nagłówek z odpowiednią klasą (do pobrania tu: <https://create.stephan-brumme.com/hash-library/>). Następnie proszę w programie dodać punkty pomiaru czasu początkowego i końcowego (jak to omówiłem w zamieszczonym w materiałach filmie) i zwiększając trudność obliczeń, wypisywać czas, jaki był potrzebny na „zgadnięcie” wyniku. Odnośnik do najnowszych wartości hash BTC: <https://www.blockchain.com/btc/blocks> P.s. *If you are not a dumb programmer you will obviously write a for loop :*)