LABORATORIUM IV
ZADANIA

**Zadanie 1 (obowiazkowe), 1 pkt.** W tym zadaniu będziemy zajmować się kryptosystemem RSA. Będziemy szyfrować teksty blokami, zamieniając każdą z liter na liczbę według poniższej tabeli.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

| R | S | T | U | V | W | X | Y | Z | ␣ |
|---|---|---|---|---|---|---|---|---|---|
| 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |

Na przykład ALA to liczba 102110.

W pliku `rsa.txt` znajduje się klucz publiczny i zaszyfrowane za jego pomocą słowa (po jednym dla każdego uczestnika zajęć). Proszę je rozszyfrować wiedząc, że zostały one (przypadkowo) podzielone na dwa podsłowa i każde z tych podsłów zostało oddzielnie zaszyfrowane.

W rozwiązaniu można wspomóc się pakietem *Mathematica* i użyć funkcji `FactorInteger` służącej do rozkładu liczby na czynniki. Pozostałe obliczenia należy wykonać przy użyciu programów w C++ (używając wcześniej napisanych funkcji).

**W rozwiązaniach następnych dwóch zadań należy użyć materiału z wykładu.**

**Zadanie 2 (obowiazkowe), 1 pkt.** Uzupełnić kod w pliku `linsolve.cpp` tak, aby całość wypisywała na standardowe wyjście wszystkie rozwiązania równania $ax = b$ w $\mathbb{Z}_n$.

W rozwiązaniu należy użyć (napisanej już wcześniej) funkcji `inverse()`, która oblicza odwrotność elementu.

Dla przykładu równanie $12x = 28$ w $\mathbb{Z}_{64}$ ma cztery rozwiązania: $13, 29, 45, 61$.

**Uwaga.** Nazwy zmiennych w C++ nie mogą kończyć się apostrofem.

**Zadanie 3 (obowiazkowe), 1 pkt.** W pliku `crt.cpp` uzupełnić kod funkcji
`bigInteger crt (bigInteger a[], bigInteger n[],bigInteger k)`
tak, aby zwracała ona najmniejsze nieujemne rozwiązanie układu kongruencji

$$\begin{cases} x \equiv a_1 \pmod{n_1} \\ x \equiv a_2 \pmod{n_2} \\ \vdots \\ x \equiv a_k \pmod{n_k} \end{cases}$$

Tablice `a[],n[]` przechowują odpowiednio współczynniki $a_i$ oraz (względnie pierwsze) moduły $n_i$. Parametr $k$ określa liczbę równań.

Dla przykładu dla poniższego układu

$$\begin{cases} x \equiv 17 \pmod{13111} \\ x \equiv 15 \pmod{225} \\ x \equiv 7 \pmod{13} \\ x \equiv 5 \pmod{19} \\ x \equiv 45 \pmod{9999991} \end{cases}$$

program powinien wypisać $1707491663256165$.

**Zadanie 4 (domowe), 1 pkt.** Napisać program, który szyfruje podany plik tekstowy za pomocą algorytmu RSA. W rozwiązaniu szyfrowanie powinno odbywać się blokami o "dużej" długości, dodatkowo aby zwiększyć bezpieczeństwo bloki powinny zawierać elementy losowości (np. ostatnie K bajtów powinno być losowane). Do rozwiązania proszę dołączyć komentarze tłumaczące działanie kodu.