Nazwisko (drukowanymi):	Imię:	Grupa dziekańska:	dn	mc	rok

Ochrona danych - ZADANIA – (3)

Uwaga: przy pisaniu poniższych programów problemem mogą być ograniczenia możliwości zmiennych co do wielkości przechowywanych liczb. Dlatego proszę stosować odpowiednie typy danych (całkowite bez znaku) i zwracać uwagę na to, czy pomieszczą one generowane wyniki.
Ponadto, nie można korzystać z gotowych funkcji (dostępnych w różnych językach programowania), które pozwalają np. na testowanie wylosowanych pod kątem ich pierwszości, generowanie kluczy lub szyfrowanie danych.

Zadanie 1 /3 pkt./
Napisz program *TestFermata*, który:
1. wylosuje nieparzystą liczbę dodatnią lub pozwoli użytkownikowi na podanie swojej liczby,
2. sprawdzi – wykorzystując test Fermata – czy liczba może być pierwsza,
3. wyświetli zaprzeczenie lub brak zaprzeczenia co do pierwszości liczby.

W programie musi być możliwość (opcja):
- określenia górnej granicy losowanej liczby (powinna ona wynosić co najmniej 10^9),
- określenia liczby prób w teście Fermata.

Zadanie 2 /3 pkt./
Napisz program *WyborKluczy*, który – wykorzystując rozszerzony algorytm Euklidesa – w oparciu o dwie liczby losowe „p" i „q", pozwoli na wygenerowanie pary kluczy: Prywatnego {e, n} i Publicznego {d, n}, takich że:
1. Liczba „e" będzie losowa,
2. Liczba „e" i (p-1)*(q-1) będą względnie pierwsze,
3. Liczba „d" będzie odwrotna do „e" modulo (p-1)*(q-1), czyli e*d = 1 mod (p-1)*(q-1),
4. Liczba „n" będzie równała się iloczynowi „p" i „q", czyli n = p*q.

Program musi:
- wykonywać obliczenia dla liczb „p" i „q" o wielkości do 10^9, tak żeby „n" mogło wynosić nawet 10^{18},
- wyświetlać wygenerowane pary kluczy i zapisywać je w plikach o nazwach – odpowiednio – „public.key" i „private.key" (liczby muszą znajdować się w oddzielnych wierszach).

Zadanie 3 /6 pkt./
Napisz program *SzyfrujRSA*, który – w oparciu o parę kluczy Prywatny {e, n} i Publiczny {d, n}, zapisanych w plikach „private.key" i „public.key" (zob. zadanie wcześniejsze) – pozwoli na:
1. zaszyfrowanie (ze wskazanym kluczem) tekstu umieszczonego w pliku „text.txt", wyświetlenie go na ekranie i zapisanie w pliku „text.enc"
2. odszyfrowanie (ze wskazanym kluczem) tekstu podanego w pliku „text.enc", wyświetlenie go na ekranie i zapisanie w pliku „text.dec"

Dla ułatwienie i ze względu na długość kluczy, program powinien:
- szyfrować tylko pliki tekstowe biorąc po dwa znaki równocześnie w ten sposób, że kody ASCII (w systemie dziesiętnym) kolejnych znaków, traktowane są jako trzy (!) cyfry jednej liczby.
  Przykładowo, tekst „KO" (kody ASCII: 75, 79) da liczbę 075079. Jeżeli w pliku będzie nieparzysta liczba znaków, to program doda do liczby otrzymanej z ostatniego znaku trzy zera (000).
- zapisać wynik szyfrowania (zawartość pliku text.enc) w postaci liczb dziesiętnych oddzielonych od siebie przecinkami, np. 1274,8723,125,29902.