UVOD

U postavci zadatka implementirano je segmentovanje, enkodovanje i dekodovanje teksta. Potrebno je proširiti postavku tako da se omogući enkriptovanje i dekriptovanje ulazne tekstualne datoteke. Pre toga je neophodno generisati javni i privatni ključ kojima će se vršiti kriptografske operacije.

Implementirani delovi algoritma:

- generisanje prostih brojeva (`generate_primes`)
- računanje najmanjeg zajedničkog sadržaoca (`calculate_lcm`)
- računanje najvećeg zajedničkog delioca (`calcultate_gcd`)
- računanje inverzne modularne multiplikacije (`inverse`)

ZADACI

1. Proširiti funkciju `make_key_pair` koja generiše javni i privatni ključ. Povratna vrednost ove funkcije su objekti klasa `PublicKey` i `PrivateKey` koji zajedno čine uređeni par RSA ključeva.

Algoritam za generisanje ključeva:

1. Izabrati dva prosta broja p i q tako da važi:
   - $n = p * q$
   - $n\_min < n < n\_max$
2. $l = NZS(p-1, q-1)$
3. Izabrati e tako da je $3 \leq e < l$ i važi da je $NZD(e, l) = 1$ (e i l su uzajamno prosti)
4. $d = e^{-1} \bmod l$ – inverzna modularna multiplikacija

2. Proširiti funkcije `encrypt` i `decrypt` implementacijom odgovarajućih kriptografskih operacija. Argumenti funckije `encrypt` su objekat klase `PublicKey` i segment teksta koji se enkriptuje. Argumenti funkcije `decrypt` su objekat klase `PrivateKey` i segment teksta koji de dekriptuje.

Eksponent koji se koristi za **enkripciju**:	Eksponent koji se koristi za **dekripciju**:
$c = m^e \bmod n$	$m = c^d \bmod n$

3. Nakon pokretanja programa na ulazu se očekuje komanda. Dozvoljene komande su:

    - `genkey` – komanda za generisanja para ključeva koja očekuje dužinu ključa u bitima (preporučena vrednost je 48b)
    - `encrypt` – komanda za enkripciju teksta, kao dodatni parametar očekuju se imena ulaznih i izlaznih datoteka (kao primer data je datoteka *input*)
    - `decrypt` – komanda za dekripciju teksta, kao dodatni parametar očekuju se imena ulaznih i izlaznih datoteka