VIGENERJEVA ŠIFRA

Funkcija Encrypt sprejme dva argumenta, besedilo, ki ga želimo zašifrirati in ključ s katerim to naredimo. Preden se začne enkripcija, se izvede funkcija nastaviKljuc, ki popravi kljuc, da sovpada z dolzino besedila. Nato gremo cez celotno besedilo, najdemo indeks vsake črke besedila in ključa in nato te dve vrednosti seštejemo po modulu 26 (toliko kot je črk v angleški abecedi). Nato pogledamo katera črka je na indeksu, ki je enak vsoti, ki smo jo dobili. Ta črka, ki se tam nahaja, je zakriptirana črka našega besedila.

Funkcija Decrypt je zelo podobna funkciji Encrypt, razlika je samo v tem, da indekse črk besedila in ključa tukaj odštejemo po modulu 26.

Funkcija FindKeyLenght spremjem en argument, in sicer kriptogram, ki ga želimo dekriptirati. Ideja je, da najde ponavljajoče “substringe” v besedilu in njihovo razdaljo. Nato gledamo deljitelj razdalj, ki se največkrat ponovi.

V funkciji poisciPonavljanje dobimo slovar, ki ima za key podniz in za value razdaljo med zaporednima podnizoma.

Nato s funkcijo Deljitelji poiščem vse deljitelje števila, v našem primeru od vsake razdalje in to dodam v slovar praFaktorji, kjer je sedaj key podniz in value razdalja zapisana z deljitelji.

Funkcija NajpogostejsiFaktorji vrne tabelo parov, kolikokrat se nek faktor pojavi.

Nato poiščemo faktor, ki je večji od 2, ki se največkrat ponovi in to je naša dobljena dolžina ključa.

Metodi se reče metoda Kasiskega[[1]](#footnote-1).

Funkcija findKey pa sprejme kriptogram in dolžino ključa, na koncu pa najde ključ kriptograma. S for zanko gremo cez celotno dolžino ključa.

S funkcijo naKateremZacnemo(c,i,dolžina) določimo indeks, kjer bomo začeli v kriptogramu in ce se dolžini ujemata (je ostanek pri deljenju enak 0), potem se dolžina odsekga poveča. Ta funkcija vrne nek del kriptograma na katerem nato izvajamo Cezarjevo šifro.

Ko imamo pripravljen odsek kriptograma, gremo s for zanko cez celotno abecedo in za vsak indeks naredimo Cezarjevo šifro, v obratno smer.

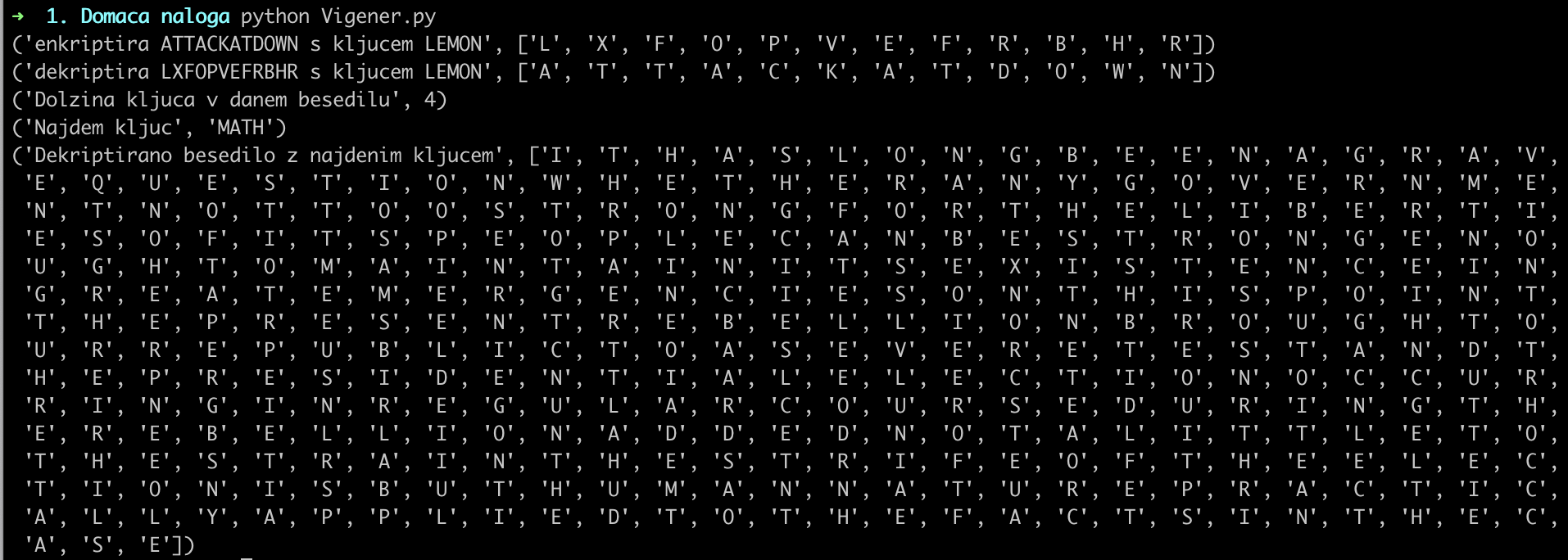
Funckija caesar zamika vse črke za določen parameter in po modulu 26. Mi v naši funkciji vstavimo za shift = - j (j je trenutna vrednosti v for zanki).

Za vsako cezarjevo “premetanko” nato izvajamo frekvenčno analizo.

Funkcija frkevenceCrk prešteje kolikokrat se neka črka pojavi v danem besedilu in deli z dolžino celotnega besedila, nato pa od tega odštejemo frekvence angleških črk, ki so podane v globalni tabeli, da dobimo vrednosti, ki predstavljajo, kako “blizu” je besedilo frekvencam crk angleške abecede. Manjša je ta razlika, boljše je.

Nato izmed vseh razlik poiščemo najmanjšo in ustrezen zamik, saj bo ravno indeks, ki ustreza zamiku predstavljal črko našega ključa. Ko pridemo čez celoten for, dobimo rešitev.

Spodaj prilagam sliko, kjer so slikane rešitve v terminalu. Za Encrypt in Decrypt sem vzela nek manjši problem, za dolžino ključa in avtomatsko najdenje ključa pa sem uporabila podan primer.



HILLOVA ŠIFRA

Funkcija Encrypt spremje dva argumenta, besedilo, ki ga želimo zakodirati in ključ s katerim bomo to naredili. Preden se lotimo kodiranja, preverimo, ce je besedilo večkratnik dolžine ključa in obe besedi spravimo v matriki.

matrikaBesedila je funkcija, ki naredi dvodimenzijonalno tabelo in gre nato čez polovico stolpcev in čez vse vrstice, da element v novi matriki dobi vrednosti indeksa na katerem se nahaja črka kriptograma. Najprej napolnimo vse vrstice in se premikamo po stolpcih naprej.

matrikaKljuča je funckija, ki naredi dvodimenzijonalno tabelo za besedo, ki jo predstavlja ključ, je 2x2 matrika, gremo cez stoplce in vrstice in napolnimo tabelo z indeksi na katerih so črke ključa.

Nato zmnožimo matriki ključa in besedila. Pazimo, da je zmnožena po modulu 26. Nastala matrika ima elemente, ki so že črke našega zakriptiranega besedila. Sedaj gremo samo še z dvema for zankama po celotni matriki in izpisujemo črke, ki so na indeksih predstavljeni z elementi matrike, po stolpcih.

Funckija Decrypt prav tako sprejme dva argumenta, kriptogram, ki ga zelimo v prvotno obliko in ključ. Funkcija je zelo podobna Encrypt, samo da pred množenjem matrik ključa in besedila, najdemo inverz matrike ključa.

Funkcija inverz vrne inverz matrike, ki je podana kot argument. Najprej izračunamo determinanto, po modulu 26. Nato najdemo inverzno število te determinante po modulu 26.

Funkcija inverzDeter najde inverz števila po modulu 26, tako, da gremo cez vse elemente in pri vsakem pogledamo, ce ima zmožek elementa in našega števila kongruentna 1. Pri tistem elementu, ki se to zgodi, je naš inverz.

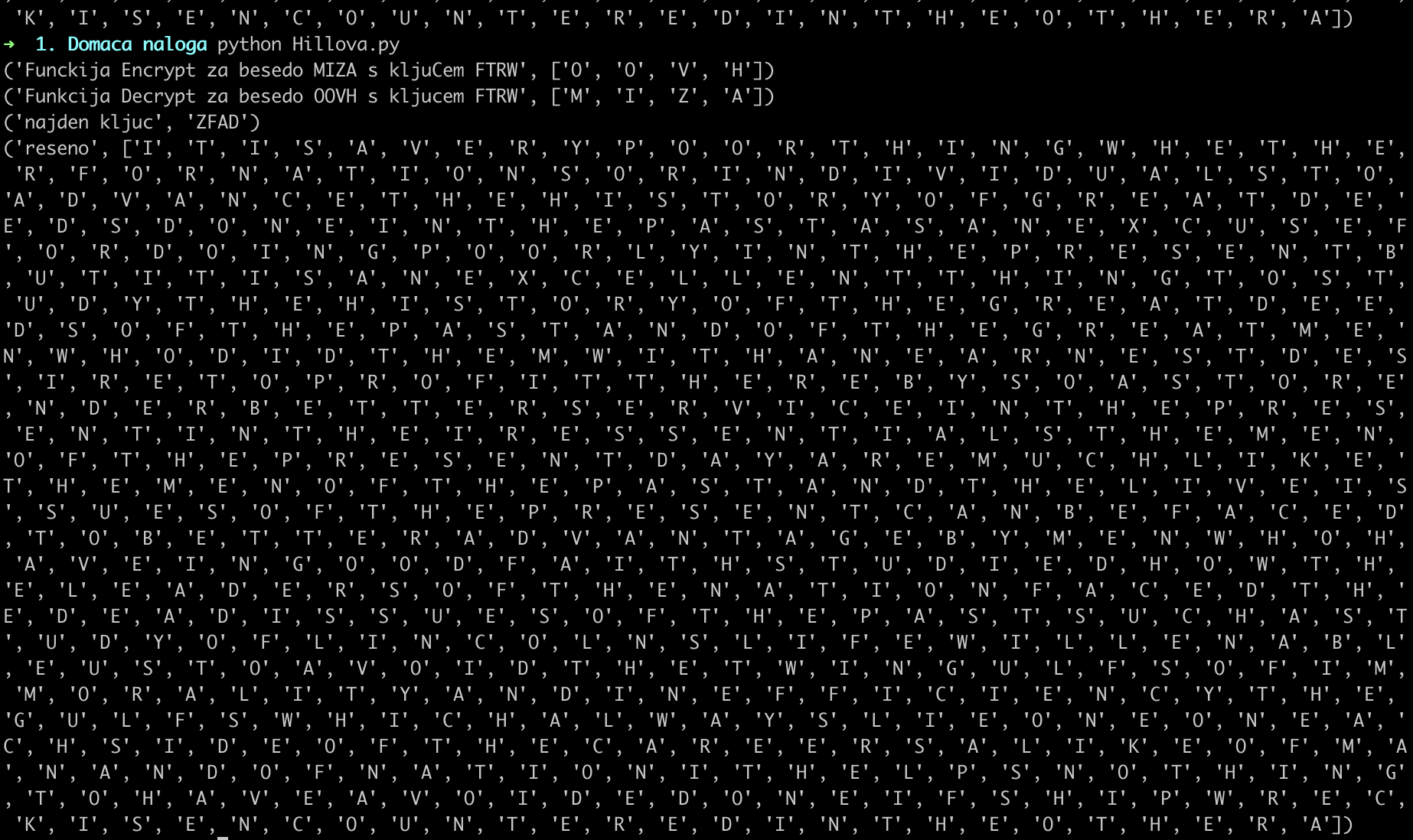
Nato še zamenjamo elemente v matriki in jih vse pomnožimo z inverzno determinanto.

Nato samo matrično zmnožimo inverzno matriko ključa z matriko besedila in nato spet po stolpcih izpisujemo pripradajoče črke.

Funkcija FindKey sprejme kriptogram za katerega želimo najdti ključ in ga nato dekriptirati. Najprej poiščemo besedi dolžine 2, ki se največkrat ponovita.

Funkcija poisciBesedo sprejme kriptogram in gre nato po celem besedilu, da najde kolikokrat se pojavijo različne besede dolžine 2. Iz slovarja izločim podatke, da dobim “substring” in število pojavitev. Nato s funkcijo najveckratPojavi dobim indeksa podnizov, ki se pojavita največkrat. Na tistih indeksih poiščem ustrzne črke.

Iz dobljenih črk naredim matriko s funkcijo matrikaKljucev. Določim najpogostejši podniz v angleški abecedi “TH” in drugi najpogostejši “HE”, ju združim in spet naredim matriko. Nato pa še njen inverz. Te dve matriki zmnožim in nato pogledam pripadajoče črke in dobimo ključ za dani kriptogram.

Spodaj prilagam sliko rezultatov.

1. <https://en.wikipedia.org/wiki/Kasiski_examination> [↑](#footnote-ref-1)