# Šifrování a dešifrování

* Autoklíč z šifrovaného textu
* **Abeceda**: česká abeceda s diakritikou (aábcčdďeéěfghiíjklmnňoópqrřsštťuúůvwxyýzž)
* **Validace otevřeného textu:** validační funkce zajišťuje, že zadaný text obsahuje pouze písmena z abecedy a případně mezery.
* **Validace klíče**: validační funkce zajišťuje, že zadaný klíč obsahuje pouze písmena abecedy a neobsahuje mezery.
* **Šifrování:** Text se šifruje tak, že ke každému písmenu původního textu se přičte odpovídající písmeno z klíče (přičemž klíč je rozšiřován o zašifrované znaky). Mezery se přepíšou do šifrovaného textu (přičemž při přepisování mezery se neposouváme v klíči)
* **Dešifrování:** Dešifrování se provádí opačně než šifrování. K dešifrování se používá původní klíč, který se rozšiřuje o zašifrované znaky, podobně jako při šifrování.

# Útok na šifru

* **Výpočet indexu koincidence (IC):** Jako první se po zadání šifrovaného textu vypočte index koincidence. Tento index slouží k určení typu šifry. Pokud je IC příliš nízký, pravděpodobně jde o polyalfabetickou šifru, což spouští útok.
* **Útok obecně:** V mém kódu je implementován slovníkový útok. V kódu je uložen slovník, který obsahuje 64 205 českých slov. Slovník se slovo od slova prochází a tyto slova zkouší jako klíč. Útok je tedy velmi efektivní, pokud je skutečně jako klíč zvoleno jedno slovo. Měl jsem také v plánu zvolit brute-force útok do délky klíče 3, ale nakonec se ukázal i slovníkový útok na krátký klíč velmi efektivní, proto jsem od brute-force útoku odstoupil.
* **Útok na šifrovaný text s mezerami:** pokud jsou v textu oddělena slova mezerami, poté je zvolen tento typ útoku. Uživatel má možnost zvolit hodnotu (0-1) o kolik se slova můžou lehce lišit (aby se vyloučila chyba ve skloňování) a kolik slov může být nenalezeno ve slovníku, ale útok považovat za úspěšný. Jelikož je slovník velmi obsáhlý, tak tento typ útoku je velice účinný i při velmi nízkých hodnotách. Problém je samozřejmě pokud v šifrovaném textu nejsou mezery. Jak jsem popsal výše, útok zkouší klíče dle slovníku a porovnává dešifrovaný text se slovy ve slovníku.
* **Útok na šifrovaný text bez mezer:** pokud šifrovaný text neobsahuje mezery, tak se zvolí trochu jiný přístup k útoku. Opět se zkouší klíče dle slovníku, nyní ale první z dešifrovaného textu vypočítáme index koincidence a pokud je vyšší než uživatelem zadaná hodnota, bere se klíč jako potencionálně správný. Poté se ale dešifrovaný text ještě porovnává se slovníkem a pokud obsahuje vyšší než uživatelem zadaný počet slov, které se skládají z více, než 4 písmen (aby se omezilo porovnávání s předložkami, spojkami, citoslovcemi a podobně), tak je šifra považována za prolomenou.

# Obecné poznatky a následné vylepšení

* Velmi často se stává, že podobný klíč, generuje velmi podobný výsledek, který je považován za správný. Z tohoto důvodu po vyhodnocení útoku jako úspěšný je uživatel ještě dotázán, zda chce projít všechny ostatní možnosti a zobrazit všechny nalezená řešení
* Na konci jsou tedy sebrána všechna možná řešení a ty se ještě porovnávají a vybírají ty nejlepší.
* U šifry bez mezer se jako nejlepší výsledek považuje takový, který začíná na slovo, které je obsažené ve slovníku a je delší než 4. Toto samozřejmě vylučuje texty, které začínají na krátká slova, ale samozřejmě se dají procházet i všechny možné řešení, které nebyly vyhodnoceny jako nejlepší.
* U šifry s mezerami se jako nejlepší výsledek bere takový, jehož první slovo delší než 4 je obsaženo ve slovníku. Samozřejmě opět lze zobrazit veškeré řešení
* Důvod, proč považuju za nejlepší výsledky ty, které začínají na slovo ve slovníku je ten, že při testování se mi často stávalo, že klíče dešifrovaly text jen částečně a to tak, že na začátku byly nesmyslné slova a později slova správná.

# Obrázky s prolomením šifry

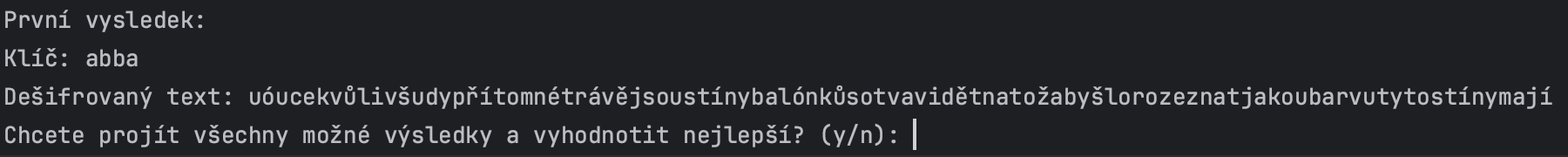
**Slovo bez mezer s klíčem jakožto slovem**

**Obsah obrázku text, snímek obrazovky, Písmo

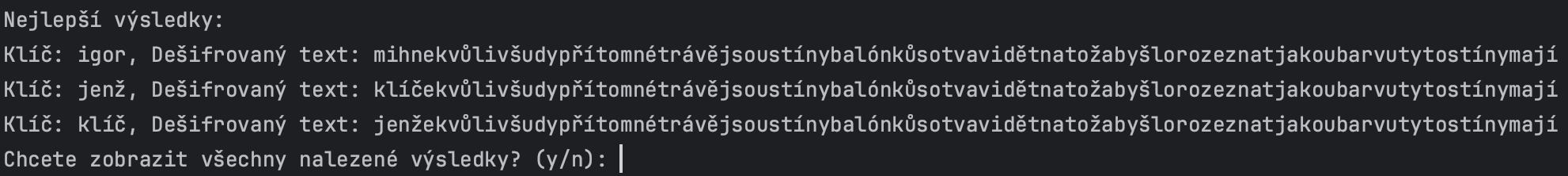
Popis byl vytvořen automaticky**

**Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky**

****

-Již zmiňovaný výsledek, kdy začátek nedává smysl, ale konec ano, pokračujeme tedy k nejlepším výsledkům



**Slovo s mezerami s klíčem jakožto slovem**

**Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky**

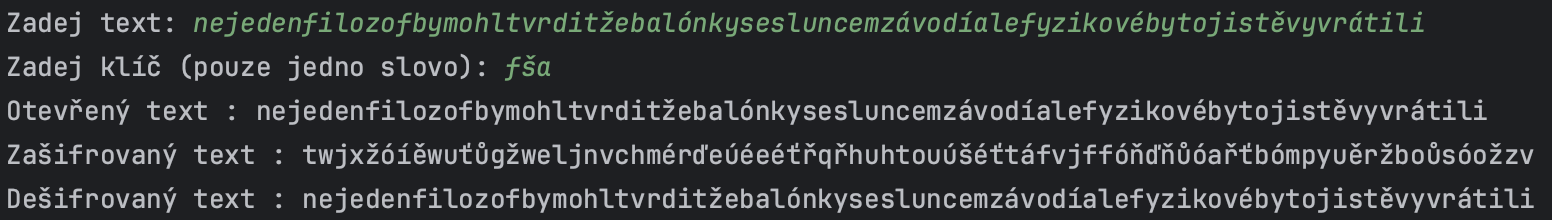
**Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky**

**Obsah obrázku text, snímek obrazovky, Písmo, černá

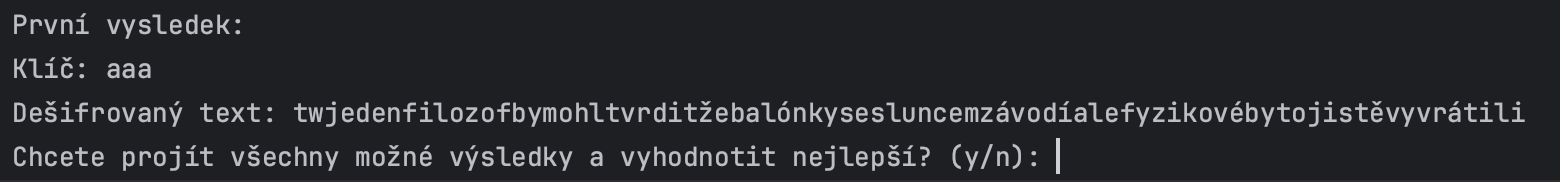
Popis byl vytvořen automaticky**

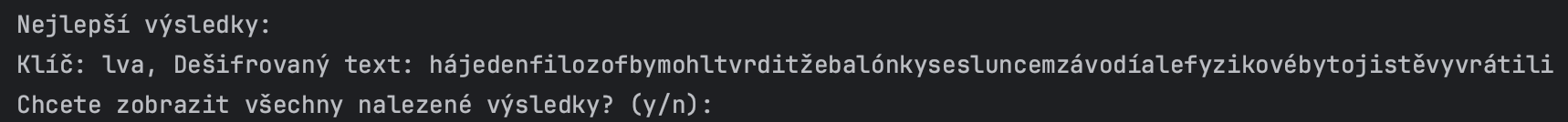
**Slovo bez mezer s „nesmyslným“ krátkým klíčem**

****

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky





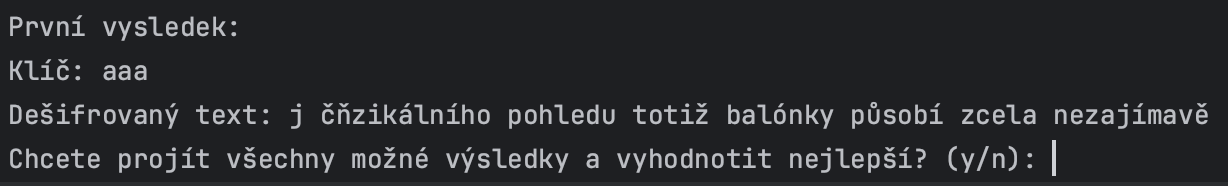
**Slovo bez mezer s „nesmyslným“ krátkým klíčem**

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky



Obsah obrázku text, snímek obrazovky, černobílá, kniha

Popis byl vytvořen automaticky

**Slovo bez mezer s „nesmyslným“ dlouhým klíčem**

-zde již výsledky tak přesné nejsou.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo, černá

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo, černobílá

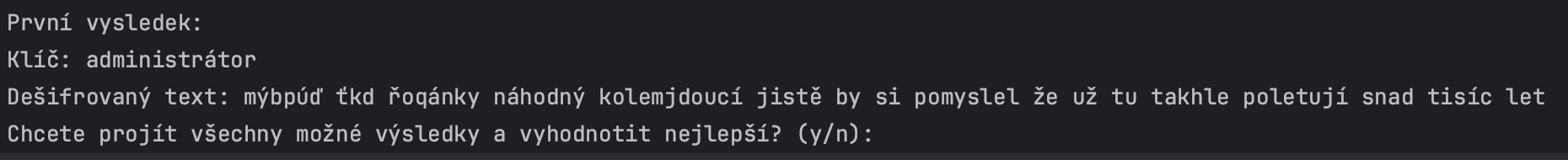
Popis byl vytvořen automaticky

**Slovo s mezerami s „nesmyslným“ dlouhým klíčem**

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, černobílá, černá

Popis byl vytvořen automaticky

**Odkaz na gitHub s README filem, který obsahuje návod ke spuštení:**  
<https://github.com/JanSkacel01/AutokeyAttack>