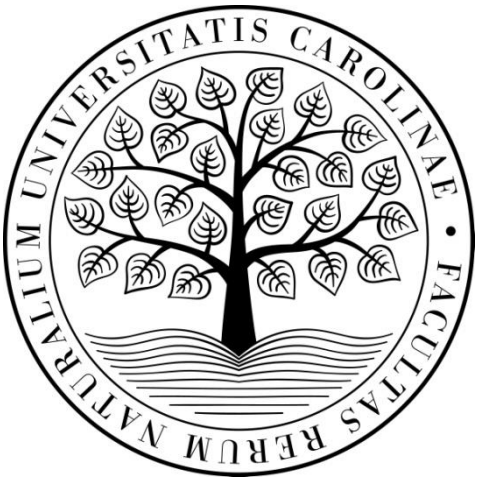
Univerzita Karlova
Přírodovědecká fakulta



Úvod do programování

Caesarova šifra

Technická zpráva

Vojtěch Schreiner
2. ročník – B-FGG

Jestřebí 2022

Zadání

109 Šifrování a dešifrování textu: Caesarova šifra

- Kategorie: Zajímavé algoritmy
- Obtížnost: 3

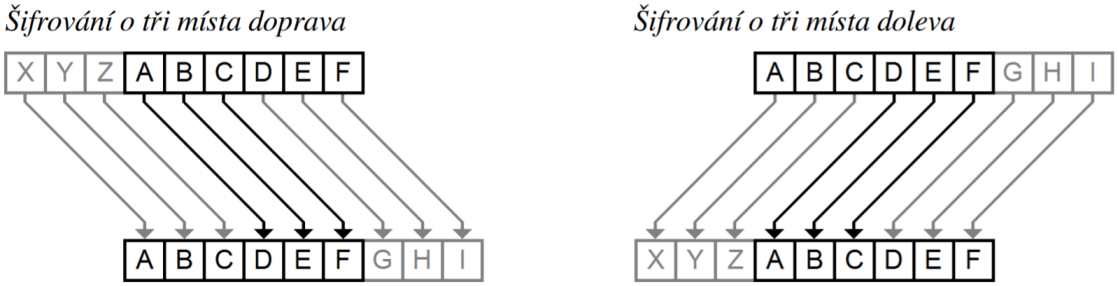
O Caesarově šifře

Caesarova šifra pochází z roku 50 př.n.l. a jedná se o klasický substituční systém (znaky otevřeného textu jsou nahrazovány jinými znaky, dle předem dohodnutého systému). Caesar tuto šifru používal i při dopisování s egyptskou královnou Kleopatrou.
Princip Caesarovy šifry: šifrování probíhalo tak, že se každý znak nahradil znakem, který je v abecedě o 3 pozice před ním. Substituční klíč tedy vypadal takto:

otevřený text: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

zašifrovaný text: X Y Z A B C D E F G H I J K L M N O P Q R S T U V W

Algoritmus pro dešifrování byl analogický: každý znak se nahradil znakem, který je o tři pozice za ním. Zároveň se jedná o příklad tzv. monoalfabetické šifry. (BITTO 2003)



Obrázek 1: Ukázka posunu o 3 místa (FAV ZČU 2021)

Monoalfabetická šifra: princip spočívá v přiřazení každému znaku abecedy jiný znak ve stejné abecedě. Počet různých klíču, které mohou být použity se rovná počtu způsobů, kterými je možné seřadit použitou abecedu. (NOVÁK 2018)

Kód

Funkce

Program provádí šifrování a dešifrovaní písmen anglické abecedy a čísel. Program posunuje velká i malá písmena anglické abecedy, pokud narazí na jiné znaky, například:

- Specifická písmena jiných abeced (*č, Ý, ů…)*
- Interpunkce (,+ . /:)
- Mezera…

Tak tyto znaky zůstanou beze změny.

Program umožňuje provádět tři úkony:

1. Šifrování: Text je zašifrován dle zadaného posunu.
2. Dešifrování: Text je možné dešifrovat pomocí známého posunu.
3. Prolomení kódu: Program aplikuje na text, všech 25 možných posunů a vypíše všechny výsledky. Uživatel si následně může vybrat řešení, které se mu zdá správné.

Vstup

Vstupní data je možno vložit dvěma způsoby:

1. Napsat přímo do konzole programu, až si to vyžádá.

2. Vložit vstup jako soubor typu .txt do složky programu. Soubor je nutno pojmenovat: *Vstup_Kod.txt*

Vstupem je tak prostý text.
U šifrování a dešifrování je třeba ještě zadat posun do konzole.
Vstup není nijak kontrolován, jelikož by nemělo být možno vložit ani do konzole ani do souboru typu .txt znak, který by nešel zpracovat a způsobil pád programu.

Výstup

Výstup dat je možný dvěma způsoby:

1. Výpis do konzole. Tento výstup je proveden vždy, není k němu potřeba nastavení uživatelem.
2. Výpis do textového souboru. Vytvoří se soubor typu .txt, ve složce s programem, který bude obsahovat stejný výstup, jaký můžete vidět v konzoly. Tuto možnost je potřeba zvolit až se Vás program optá.
    a. Pro výsledek šifrování se vytvoří soubor: Vystup_Sifrovani.txt
    b. Pro výsledek dešifrování: Vystup_Desifrovani.txt
    c. Pro všechny možnosti, pokud neznáme posun: Vystup_Prolamovace_Kodu.txt

Provedení posunu

Posun je v kódu proveden následujícím příkazem:

(jako příklad uvedeno šifrování velkých písmen)

```
for i in range(len(input)):
  ave = input[i]
  if 97<=ord(ave)<=122:
      exitus += chr((ord(ave)+ trabea - 97) %26 + 97)
```

- První řádek: Počet provedení je roven počtu znaků ve vstupních datech.
- Druhý řádek: Ze vstupních dat vybereme jen jeden znak a uložíme do proměnné.
- Třetí řádek: Ověříme, jestli je vybraný znak velké písmeno anglické abecedy. Pomocí příkazu *ord* zjistíme hodnotu znaku v Unicode a zjistíme, jestli se nachází v intervalu 97 *(Unicode pro A)* a 122 *(Unicode pro Z)*. Unicode intervaly lze zjistit příkazy *print(ord("A"))* a *print(ord("Z"))*.
- Čtvrtý řádek:
    - 1) Vezmeme Unicode znaku uloženého v proměnné a přičteme hodnotu posunu (*trabea = posun (v latině)*)
    - 2) Odečteme 97, toto je hodnota, na které začíná Unicode pro velká písmena, výsledkem je tedy číslo v intervalu <0, 25>
    - 3) Vydělíme číslem 26 *(počet písmen anglické abecedy)* a dále pracujeme se zbytkem tohoto dělení. Tímto ošetříme posuny větší než 25, které již nevedou k žádným novým výsledkům.
    - 4) Přičteme zpět 97, získáme tak Unicode nového znaku.
    - 5) Pomocí *chr* převedeme Unicode na znak, ten následně přidáme do proměnné *exitus*.

Při dekódování se mění znaménko u posunu z plus (+) na minus (-).

```
      exitus += chr((ord(ave)- trabea - 97) %26 + 97)
```

Provedení posunu u čísel

Posun u čísel se od posunu písmen liší hodnotou čísla, kterou dělíme, abychom nedostávali opakované výsledky místo čísla 26 použijeme 10, jelikož máme jenom 10 čísel na rozdíl od 26 znaků anglické abecedy.

```
    elif 48<=ord(ave)<=57:
        exitus += chr((ord(ave)+ trabea - 48) %10 + 48)
```

Znak, který nechceme měnit

Pokud nechceme znak měnit přidáme pouze znak do výsledku.

```
    else:
        exitus += ave
```

Uživatelská dokumentace

Vážený uživateli, do rukou se Vám dostal program umožňující šifrování a dešifrování textu v Caesarově šifře. Tento program šifruje/dešifruje čísla a malá a velká písmena, ostatní znaky zůstávají beze změny. Tato dokumentace Vás provede použitím tohoto programu.

Prvním úkonem, co musíte udělat po spuštění je určení, jak vložíte výchozí text. Konkrétně jestli chcete vkládat text ve formátu .txt.

- Pokud zvolíte ano (Y): Musíte mít ve složce programu vstupní soubor pojmenovaný: Vstup_Kod.txt.
- Pokud zvolíte ne (N): Budete zadávat vstupní text do konzole.

Dalším krokem po nahrání vstupu je rozhodnutí, jestli budete šifrovat nebo dešifrovat. Zde máte tři možnosti:

- Šifrování (E): Pokud zvolíte šifrování program se Vás následně zeptá, jestli budete chtít nový zašifrovaný soubor uložit jako textový soubor a na číselnou hodnotu posunu.
- Dešifrování (D): Pokud zvolíte dešifrování program se Vás následně zeptá, jestli budete chtít nový dešifrovaný soubor uložit jako textový soubor a na číselnou hodnotu posunu použitého při šifrování.
- Prolomení kódu (B): Tuto funkci můžete použít, pokud neznáte hodnotu posunu při šifrování. Program vypíše výsledky pro všechny možné posuny a umožní vám uložit výstup jako textový soubor.

Výstup je vždy vypsán do konzole. Výstupy v podobě textových souborů jsou uloženy do složky programu.

Zdroje

1. BITTO O. (2003): Historie kryptologie. Dostupné z: https://www.fi.muni.cz/usr/jkucera/pv109/2003/xbitto.htm#_Nejstar%C5%A1%C3%AD_%C5%A1ifrovac%C3%AD_algoritmy
2. FAV ZČU (2021): Caesarova šifra. Dostupné z: https://pilsprog.fav.zcu.cz/tasks/2021.pdf
3. NOVÁK M. (2018): Jednoduchá monoalfabetická šifra. Dostupné z: https://wikisofia.cz/wiki/Jednoduch%C3%A1_monoalfabetick%C3%A1_%C5%A1ifra