

České vysoké učení technické v Praze FIT

# Programování v Pythonu

Jiří Znamenáček

*Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.*

*Praha & EU: Investujeme do vaší budoucnosti*



# Procvičování 1

1. Napište program pro *Césarovu šifru*. Tato šifra je založena na posouvání písmen o 3 pozice v abecedě (tj. a→d, m→p, z→c). Program by měl obsahovat funkce jak pro zašifrování, tak pro dešifrování takové šifry. Tyto funkce by měly brát řetězec a vrátit jeho zašifrovanou/dešifrovanou podobu.

**[–] řešení ([procvicovani/caesar.1a.py](#))**

```
import string

ring = string.ascii_uppercase

def cipher(text):
    """Zašifruj zadaný ASCII-text."""

    txt = ''
    key = ring[3:] + ring[:3]
    for char in text.upper():
        if char in key:
            txt += key[ ring.index(char) ]
        else:
            txt += char

    return txt

def decipher(text):
    """Odšifruj zadaný ASCII-text."""

    txt = ''
    key = ring[3:] + ring[:3]
    for char in text.upper():
        if char in key:
            txt += ring[ key.index(char) ]
        else:
            txt += char

    return txt

# test
txt = cipher('Ahoj, jak to jde? Z')
print( txt )
print( decipher(txt) )
```

**[-] řešení ([procvicovani/caesar.1b.py](#))**

```
import string

ring = string.ascii_uppercase
ring_len = len(ring)

def cipher(text):
    """Zašifruj zadaný ASCII-text."""

    txt = ''
    for char in text.upper():
        if char in ring:
            txt += ring[ (ring.index(char) + 3) % ring_len]
        else:
            txt += char

    return txt

def decipher(text):
    """Odšifruj zadaný ASCII-text."""

    txt = ''
    for char in text.upper():
        if char in ring:
            txt += ring[ (ring.index(char) - 3) % ring_len]
        else:
            txt += char

    return txt

# test
txt = cipher('Ahoj, jak to jde? Z')
print( txt )
print( decipher(txt) )
```

**2.** Upravte předchozí program, aby umožňoval posouvání písmen o jiné hodnoty než 3.

**[-] řešení ([procvicovani/caesar.2a.py](#))**

```
import string

ring = string.ascii_uppercase

def cipher(text, shift):
    """Zašifruj zadaný ASCII-text uvedeným posunem."""

    txt = ''
    key = ring[shift:] + ring[:shift]
    for char in text.upper():
        if char in key:
            txt += key[ ring.index(char) ]
        else:
            txt += char

    return txt

def decipher(text, shift):
    """Odšifruj zadaný ASCII-text uvedeným posunem."""

    txt = ''
    key = ring[shift:] + ring[:shift]
    for char in text.upper():
        if char in key:
            txt += ring[ key.index(char) ]
        else:
            txt += char

    return txt

# test
txt = cipher('Ahoj, jak to jde? Z', -1)
print( txt )
print( decipher(txt, -1) )
```

**[-] řešení ([procvicovani/caesar.2b.py](#))**

```
import string

ring = string.ascii_uppercase
ring_len = len(ring)

def cipher(text, shift):
    """Zašifruj zadaný ASCII-text uvedeným posunem."""

    txt = ''
    for char in text.upper():
        if char in ring:
            txt += ring[ (ring.index(char) + shift) %
ring_len]
        else:
            txt += char

    return txt

def decipher(text, shift):
    """Odšifruj zadaný ASCII-text uvedeným posunem."""

    txt = ''
    for char in text.upper():
        if char in ring:
            txt += ring[ (ring.index(char) - shift) %
ring_len]
        else:
            txt += char

    return txt

# test
txt = cipher('Ahoj, jak to jde? Z', -1)
print( txt )
print( decipher(txt, -1) )
```

**3.** Napište program, který z textového souboru přečte postupně všechny řádky a pro každou vypíše součet čísel, která se na této řádce vyskytují. Na konci program navíc vypíše součet všech čísel přes celý soubor.

**[-] nápověda**

float()

**[-] řešení ([procvicovani/cisla.1.py](#))**

```
lines = []
with open("cisla.1.txt", mode="r", encoding="utf-8") as f:
    lines = f.readlines()

celkem = 0
for line in lines:
    radka = 0
    for num in line.split():
        radka += float(num)
    celkem += radka
    print( "{0:^6} : {1}".format(radka, line), end=' ' )
print(celkem)
```

4. Upravte předchozí program, aby fungoval i pro čísla libovolně rozházená mezi jiným textem. Přitom uvažujeme pouze samostatně stojící čísla (tj. oddělená od okolního textu mezerou).

**[-] nápověda**

float(), ValueError

**[-] řešení ([procvicovani/cisla.2.py](#))**

```
lines = []
with open("cisla.2.txt", mode="r", encoding="utf-8") as f:
    lines = f.readlines()

celkem = 0
for line in lines:
    radka = 0
    for num in line.split():
        try:
            radka += float(num)
        except ValueError:
            pass
    celkem += radka
    print( "{0:^6} : {1}".format(radka, line), end=' ' )
print(celkem)
```

5. Napište program, který přečte soubor v kódování UTF-8 a vypíše všechny znaky, které nejsou součástí ASCII-části unicodové tabulky. Každý takový znak bude vypsán pouze jednou.

**[-] nápověda**

ord(znak) > 127

**[-] řešení ([procvicovani/znaky.py](#))**

```
text = ''
with open('hasek-svejk.txt', mode='r', encoding='utf-8') as f:
    text = f.read()

znaky = { znak for znak in text if ord(znak) > 127 }
print(znaky)
```

---

Autorem původního zadání u příkladů 1-3 a 5 je Bedřich Košata.