

Przedmiot: Bezpieczeństwo Sieci Komputerowych	Prowadzący: mgr inż Maciej Brzozowski
Student: Krzysztof Nowak	Data: 30.03.2011 r.

Dokumentacja do algorytmów kryptograficznych z listy nr. 1

Wszystkie poniższe algorytmy zostały napisane w języku Python w wersji 2.7.

Większość programów ma podobny schemat wywołania: `python [nazwa.py] [tryb] [n lub klucz] [wiadomosc]`

Przy czym:

**[tryb]** – “c” lub “d” w zależności czy interesuje nas szyfrowanie (cipher) czy deszyfrowanie (decipher)  
**[n lub klucz]** – w algorytmie railfence, liczba naturalna, w przestawieniach macierzowych permutacja np. postaci “1-3-4-5-2” lub słowo np. “CONVENIENCE”.

Wyjątkiem jest szyfr cezara gdzie należy kolejno podać dwie liczby k0, k1.

**[wiadomosc]** – ciąg znaków, najlepiej ujęty w cudzysłowiu: “ala ma kota”

## 1. Rail fence:

Przykładowe wywołanie:

```
python railfence.py c 6 "Tomorrow's lottery numbers are 6 23 13 55 7"
```

```

alice@wonderland: ~/Projekty/BSK
File Edit View Search Terminal Help
alice@wonderland:~/Projekty/BSK$ python railfence.py c 5 "How do we know if she's a w
itch ? Let's build a bridge out of her."
Hefa?bburow i      u rote.w k sswhLsiai h onwh'ice'l deo doettdgf
alice@wonderland:~/Projekty/BSK$ python railfence.py d 5 "Hefa?bburow i      u rote.w
k sswhLsiai h onwh'ice'l deo doettdgf"
How do we know if she's a witch ? Let's build a bridge out of her.
alice@wonderland:~/Projekty/BSK$ |

```

## 2. Przestawienia macierzowe:

a)

Przykładowe wywołanie:

```
python matrix_perm.py c 3-4-1-5-2 "So mr. Bond... we meet again."
```

```

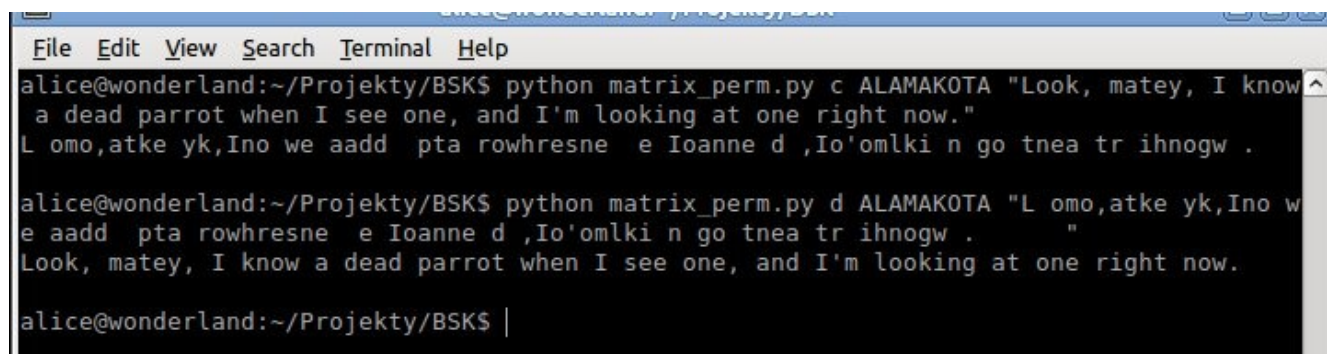
alice@wonderland: ~/Projekty/BSK
File Edit View Search Terminal Help
alice@wonderland:~/Projekty/BSK$ python matrix_perm.py c 5-3-2-1-4 "I wish to complain
about this parrot what I purchased not half an hour ago from this very boutique."
sw Ii t holmocpania uobt ihtsorapraw thpI t acruhndes a tohn flaroh u ga o orfm ihts r
evyiuobt euq.
alice@wonderland:~/Projekty/BSK$ python matrix_perm.py d 5-3-2-1-4 "sw Ii t holmocpania
uobt ihtsorapraw thpI t acruhndes a tohn flaroh u ga o orfm ihts revyiuobt euq."
I wish to complain about this parrot what I purchased not half an hour ago from this ve
ry boutique.
alice@wonderland:~/Projekty/BSK$ |

```

b)

Przykładowe wywołanie:

```
python matrix_perm.py c JOHNNY "A strange game. The only winning move is not to play. How about a nice game of chess ?"
```



```
alice@wonderland:~/Projekty/BSK$ python matrix_perm.py c ALAMAKOTA "Look, matey, I know a dead parrot when I see one, and I'm looking at one right now."
L omo,atke yk,Ino we aadd pta rowhresne e Ioanne d ,Io'omlki n go tnea tr ihnogw .

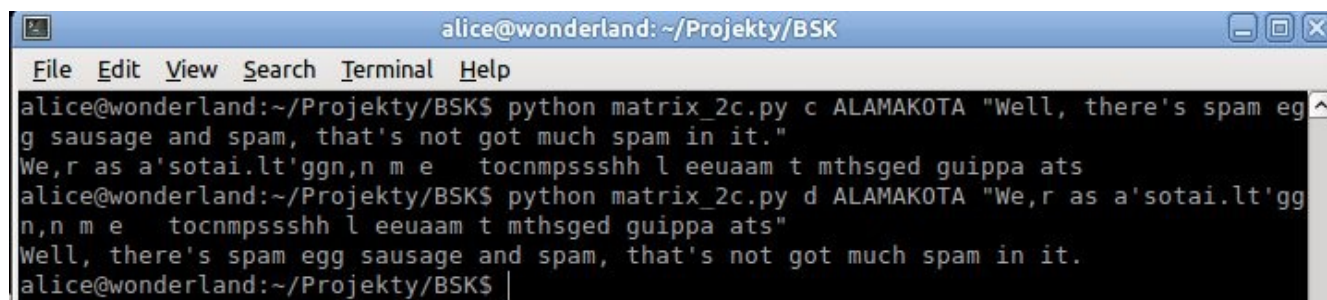
alice@wonderland:~/Projekty/BSK$ python matrix_perm.py d ALAMAKOTA "L omo,atke yk,Ino w
e aadd pta rowhresne e Ioanne d ,Io'omlki n go tnea tr ihnogw . "
Look, matey, I know a dead parrot when I see one, and I'm looking at one right now.

alice@wonderland:~/Projekty/BSK$ |
```

c)

Przykładowe wywołanie:

```
python matrix_2c.py c NECRONOMICON 'This... is my boomstick! The twelve-gauge double-barreled Remington. Shop smart. Shop S-Mart!'
```



```
alice@wonderland:~/Projekty/BSK$ python matrix_2c.py c ALAMAKOTA "Well, there's spam egg sausage and spam, that's not got much spam in it."
We,r as a'sotai.lt'ggn,n m e tocnmpssshh l eeuaam t mthsged guippa ats
alice@wonderland:~/Projekty/BSK$ python matrix_2c.py d ALAMAKOTA "We,r as a'sotai.lt'gg
n,n m e tocnmpssshh l eeuaam t mthsged guippa ats"
Well, there's spam egg sausage and spam, that's not got much spam in it.

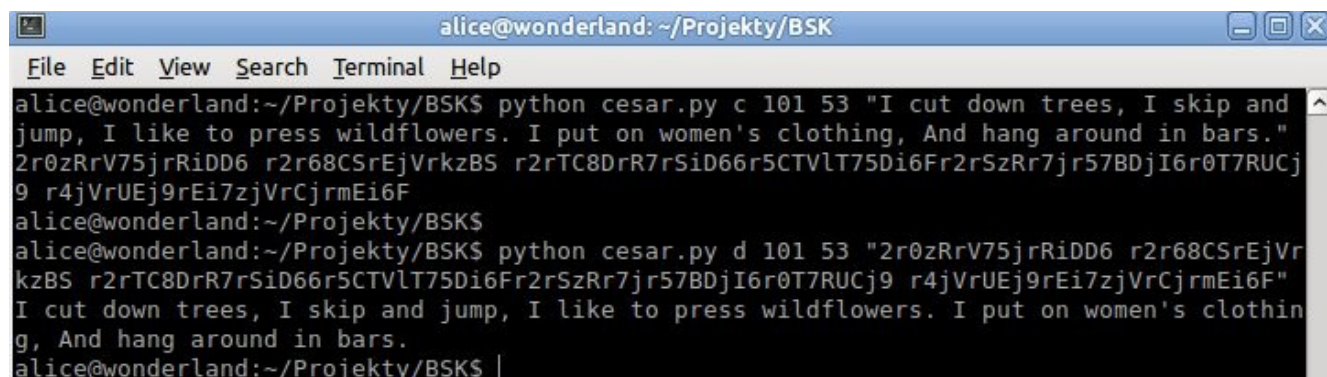
alice@wonderland:~/Projekty/BSK$ |
```

### 3. Szyfr Cezara

W odróżnieniu od standardowego szyfru, nie posługuję się kodami ascii, a definiuję własny alfabet, któremu potem przyporządkowuje indeksy w hash-mapie (aby móc się odwoływać wprost do indeksu danego znaku litery w szybkim czasie).

Przykładowe Wywołanie:

```
python cesar.py c 31 101 'I am running out of movie references.'
```



```
alice@wonderland:~/Projekty/BSK$ python cesar.py c 101 53 "I cut down trees, I skip and jump, I like to press wildflowers. I put on women's clothing, And hang around in bars."
2r0zRrV75jrRiDD6 r2r68CSrEjVrkzBS r2rTC8DrR7rSiD66r5CTVLT75Di6Fr2rSzRr7jr57BDjI6r0T7RUCj9 r4jVrUEj9rEi7zjVrCjrmEi6F
alice@wonderland:~/Projekty/BSK$
alice@wonderland:~/Projekty/BSK$ python cesar.py d 101 53 "2r0zRrV75jrRiDD6 r2r68CSrEjVrkzBS r2rTC8DrR7rSiD66r5CTVLT75Di6Fr2rSzRr7jr57BDjI6r0T7RUCj9 r4jVrUEj9rEi7zjVrCjrmEi6F"
I cut down trees, I skip and jump, I like to press wildflowers. I put on women's clothing, And hang around in bars.

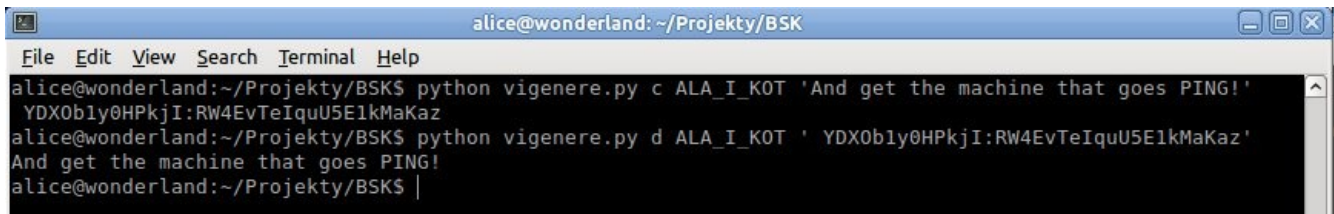
alice@wonderland:~/Projekty/BSK$ |
```

#### 4. Tablica Vigenera.

Jak wyżej, w programie zdefiniowany jest własny alfabet, wobec czego dozwolone są znaki " ! , . " itp.

Przykładowe Wywołanie:

```
python vigenere.py c CONVENIENCE "Ala ma kota."
```

A screenshot of a terminal window titled 'alice@wonderland: ~/Projekty/BSK'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the following commands and output:

```
alice@wonderland:~/Projekty/BSK$ python vigenere.py c ALA_I_KOT 'And get the machine that goes PING!'
YDX0bly0HPkjI:RW4EvTeIquU5E1kMaKaz
alice@wonderland:~/Projekty/BSK$ python vigenere.py d ALA_I_KOT ' YDX0bly0HPkjI:RW4EvTeIquU5E1kMaKaz'
And get the machine that goes PING!
alice@wonderland:~/Projekty/BSK$ |
```