Konferencja „Informatyka realnie”

Radom, 11 października 2017 roku

Tajemne przekazy

czyli o metodach informatycznych i szyfrowaniu

Opracowanie: Agnieszka Samulska

OEiiZK Ośrodek Edukacji Informatycznej i Zastosowań Komputerów w Warszawie

GA-DE-RY-PO-LU-KI

- MA-LI-NO-WE-BU-TY
- MO-TY-LE-CU-DA-KI
- NO-WE-BU-TY-LI-SA
- RE-GU-LA-MI-NO-WY
- KO-NI-EC-MA-TU-RY

ALA MA KOTA

GA-DE-RY-PO-LU-KI

GUG MG IPTG

ALA MA KOTA

NO-WE-BU-TY-LI-SA

SIS MS KNYS

Klucz:

- 12 liter
- para spółgłoska-samogłoska
- wszystkie samogłoski

Tablica kodów

	1	3	5	7	9
2	A	B	C	D	E
4	F	G	H	I	J
6	K	L	Ł	M	N
8	O	P	R	S	T
0	U	W	Y	Z	odstęp

ALA MA KOTA

12 36 12 90 76 12 90 16 18 98 12

123612907612901618981 2

Tablica kodów

- 1 papiloty
- 2 preparat
- 3 paralela
- 4 pomidory

- A 38 18 76 74 72 18 58 50
- B 38 12 58 12 36 92 36 12
- C 38 12 38 74 36 18 98 50
- D 38 58 92 38 12 58 12 98

Tablica kodów

- 1 papiloty
- 2 preparat
- 3 paralela
- 4 pomidory

- A 38 18 76 74 72 18 58 50
- B 38 12 58 12 36 92 36 12
- C 38 12 38 74 36 18 98 50
- D 38 58 92 38 12 58 12 98

1C 2D 3B 4A

Zaszyfrowana wiadomość

SMZSY_KFZOOZS
YDRBTOOKAGWR
AŁNO_Y_AO!D

Zaszyfrowana wiadomość

SMZSY_KFZOOZSYDRBTOOKAGWRAŁNO_Y_AO!D

S	M	Z	S	Y	_
K	F	Z	O	O	Z
S	Y	D	R	B	T
O	O	K	A	G	W
R	A	Ł	N	O	_
Y	_	A	O	!	D

Zaszyfrowana wiadomość

SMZSY_KFZOOZSYDRBTOOKAGWRAŁNO_Y_AO!D

S	M	Z	S	Y	_
K	F	Z	O	O	Z
S	Y	D	R	B	T
O	O	K	A	G	W
R	A	Ł	N	O	_
Y	_	A	O	!	D

Zaszyfrowana wiadomość

SMZSY_KFZOOZSYDRBTOOKAGWRAŁNO_Y_AO!D

S	M	Z	S	Y	_
K	F	Z	O	O	Z
S	Y	D	R	B	T
O	O	K	A	G	W
R	A	Ł	N	O	_
Y	_	A	O	!	D

Zaszyfrowana wiadomość

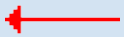
SMZSY_KFZOOZSYDRBTOOKAGWRAŁNO_Y_AO!D

S	M	Z	S	Y	_
K	F	Z	O	O	Z
S	Y	D	R	B	T
O	O	K	A	G	W
R	A	Ł	N	O	_
Y	_	A	O	!	D

Zaszyfrowana wiadomość

SMZSY_KFZOOZSYDRBTOOKAGWRAŁNO_Y_AO!D

S	M	Z	S	Y	_
K	F	Z	O	O	Z
S	Y	D	R	B	T
O	O	K	A	G	W
R	A	Ł	N	O	_
Y	_	A	O	!	D



Zaszyfrowana wiadomość

1	2	3	7	4	1
4	5	6	8	5	2
7	8	9	9	6	3
3	6	9	6	8	7
2	5	8	9	5	4
1	4	7	3	2	1

Zaszyfrowana wiadomość

1	2	3	7	4	1
4	5	6	8	5	2
7	8	9	9.	6	3
3	9.	6	6	8	7
2	5	8	6	5	4
1	4	7	3	2	1

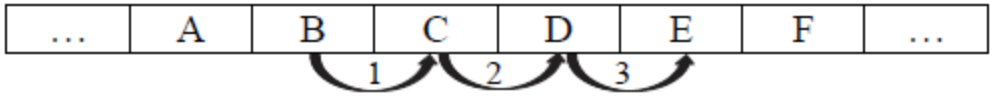
Zaszyfrowana wiadomość

szyfr przestawieniowy
tekst jawny i szyfrogram – anagramy
austriacki pułkownik
Eduard Fleissner von Wostrowitz

Szyfr Cezara

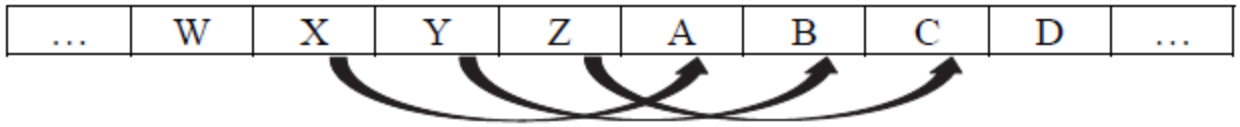
Zadanie 6. Szyfr Cezara

Podstawieniowy szyfr Cezara z przesunięciem (kluczem) *k* polega na zastąpieniu każdego znaku jawnego znakiem leżącym w alfabecie o *k* pozycji w prawo od zastępowanego znaku. Przykład: znak ‘B’ po zakodowaniu kluczem *k*=3 zastąpiony zostanie znakiem ‘E’.



Przy szyfrowaniu znaku należy postępować w sposób cykliczny, to znaczy, jeżeli znak nie posiada w alfabecie następnika przesuniętego o *k* pozycji, to alfabet „zawija się" i za literą Z następuje znów litera A.

Przykład: jawny znak ‘X’ po zakodowaniu kluczem *k*=3 zastąpiony zostanie znakiem ‘A’, znak ‘Y’ – znakiem ‘B’, natomiast ‘Z’ – znakiem ‘C’.



W tym zadaniu rozpatrujemy tylko słowa zbudowane z wielkich liter alfabetu angielskiego (o kodach ASCII odpowiednio od 65 do 90), o długościach nie większych niż 30 znaków.

Szyfr Cezara

Zadanie 6.1. (0–3)

W pliku *dane_6_1.txt* znajduje się 100 słów. Słowa umieszczono w osobnych wierszach.

Fragment pliku *dane_6_1.txt*:

```
INTERPRETOWANIE
ROZWESELANIE
KONSERWOWANIE
```

Napisz program, który **zaszyfruje** słowa z pliku *dane_6_1.txt* z użyciem klucza $k = 107$. Wynik zapisz do pliku *wyniki_6_1.txt*, każde słowo w osobnym wierszu, w porządku odpowiadającym kolejności słów z pliku z danymi.

Uwaga:
Dla pierwszego słowa z pliku *dane_6_1.txt* (INTERPRETOWANIE) wynikiem jest słowo LQWHUSUHWRZDQLH.

```
def szyfr(napis,klucz):
    pom = ""
    for e in napis:
        pom += chr((ord(e)-ord('A')+klucz)%26+ord('A'))
    return pom
```

Szyfr Cezara

Zadanie 6.2. (0–4)

W pliku *dane_6_2.txt* zapisano 3 000 szyfrogramów i odpowiadające im klucze szyfrujące. W każdym wierszu znajduje się jeden szyfrogram (zaszyfrowane słowo) i po pojedynczym znaku odstępu odpowiadający mu klucz (maksymalnie czterocyfrowa liczba).

Fragment pliku *dane_6_2.txt*:

```
BCYKUNCM 1718
YFOGNSKGYW 7580
WARDA 9334
```

Napisz program, który **odszyfruje** słowa zaszyfrowane podanymi kluczami. Wynik zapisz w pliku *wyniki_6_2.txt*: każde odszyfrowane słowo w osobnym wierszu, w porządku odpowiadającym kolejności szyfrogramów z pliku z danymi.

Uwaga:
Dla pierwszego szyfrogramu z pliku *dane_6_2.txt* (BCYKUNCM) wynikiem jest słowo ZAWISLAK.

```
def deszyfr(napis,klucz):
    return szyfr(napis,26-klucz)
```

Szyfr Cezara

Zadanie 6.3. (0–5)

W pliku *dane_6_3.txt* zapisano 3 000 par słów, po jednej parze w wierszu, oddzielonych pojedynczym znakiem odstępu. Drugie słowo w każdej parze jest szyfrogramem pierwszego z nieznanym kluczem.

Niektóre szyfrogramy są błędne, co oznacza, że niektóre litery w słowie zakodowano z różnymi przesunięciami. Słowo ma zawsze tę samą długość co odpowiadający mu szyfrogram.

Fragment pliku *dane_6_3.txt*:

```
ZAWISLAK EFBNXQFP
KRASZEWSKI XENFMRJFXV
```

Napisz program, który wyszuka i wypisze te słowa z pliku *dane_6_3.txt*, które błędnie zaszyfrowano. Wynik zapisz w pliku *wyniki_6_3.txt*: każde słowo w osobnym wierszu, w porządku odpowiadającym kolejności tych słów z pliku z danymi.

Uwaga:
Pierwsze słowo w pliku wynikowym to `SMIGIELSKI`.

```
def czy_blad(napis,napis2):
    klucz = ord(napis2[0])-ord(napis[0])
    for i in range(len(napis)):
        if szyfr(napis[i],klucz)!=napis2[i]:
            return True
    return False
```

Szyfr Cezara

```
def zad61():
    plik = open('dane_6_1.txt', 'r')
    wynik = open('wynik61.txt','w')
    for wiersz in plik:
        wynik.write (szyfr(wiersz,107)+"\n")
    plik.close()
    wynik.close()
```

```
def zad62():
    plik = open('dane_6_2.txt', 'r')
    wynik = open('wynik62.txt','w')
    for wiersz in plik:
        napis,klucz=wiersz.split()
        wynik.write (deszyfr(napis,int(klucz))+"\n")
    plik.close()
    wynik.close()
```

```
def zad63():
    plik = open('dane_6_3.txt', 'r')
    wynik = open('wynik63.txt','w')
    for wiersz in plik:
        napis,napis2=wiersz.split()
        if czy_blad(napis,napis2):
            wynik.write (napis+"\n")
    plik.close()
    wynik.close()
```

Bezpieczeństwo danych

Sprawdzanie, czy połączenie z daną stroną jest bezpieczne

Aby dowiedzieć się, czy można bezpiecznie otworzyć daną stronę, możesz sprawdzić informacje o jej zabezpieczeniach. Jeśli nie można wejść na stronę bezpiecznie lub z zachowaniem prywatności, Chrome wyświetla ostrzeżenie.

1. Otwórz stronę w Chrome na komputerze.
2. Aby sprawdzić zabezpieczenia strony, popatrz na symbol stanu bezpieczeństwa na lewo od adresu internetowego.
   - Bezpieczna
   - Informacje lub Niezabezpieczona
   - Niezabezpieczona lub Niebezpieczna
3. Aby zobaczyć szczegóły i uprawnienia strony, kliknij ikonę. Na górze panelu zobaczysz podsumowanie informacji o stopniu prywatności połączenia w Chrome.

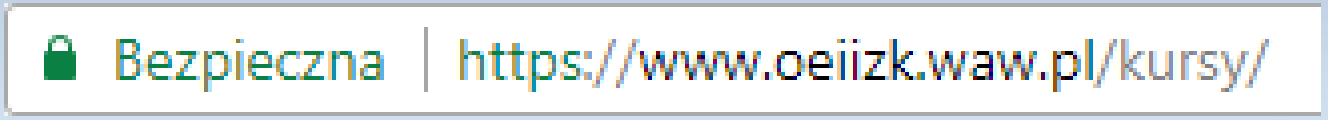
Znaczenie poszczególnych symboli bezpieczeństwa

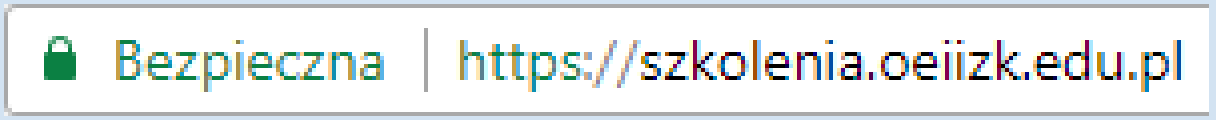
Symbole informują o tym, na ile bezpieczne jest wchodzenie na daną stronę i korzystanie z niej. Wskazują, czy strona ma certyfikat bezpieczeństwa, czy Chrome mu ufa oraz czy połączenie ze stroną jest prywatne.

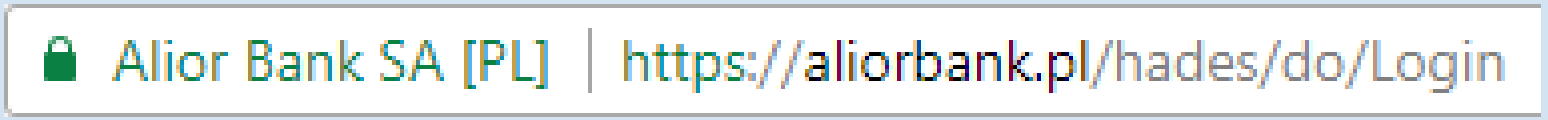
Bezpieczna

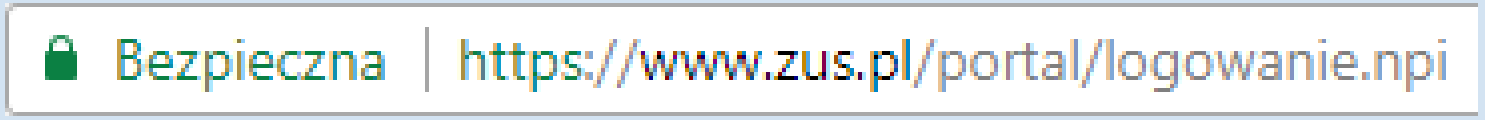
> Informacje przesyłane i odbierane na tej stronie są prywatne.
>
> Nawet gdy ta ikona jest widoczna, zawsze zachowuj ostrożność przy udostępnianiu informacji prywatnych. Sprawdź na pasku adresu, czy jesteś na właściwej stronie.

Bezpieczeństwo danych

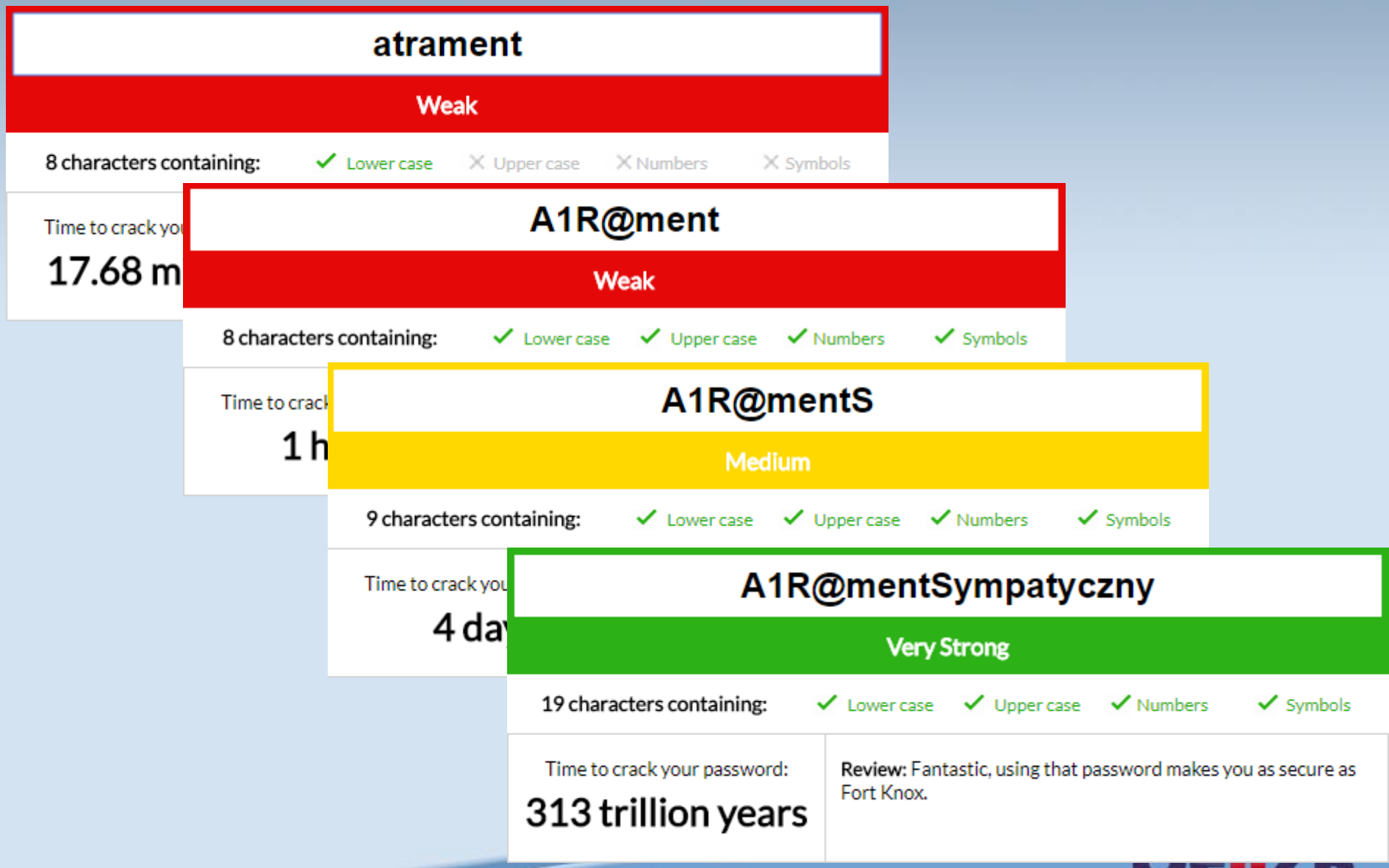








Hasło


atrament
Weak
8 characters containing:
Lower case
Upper case
Numbers
Symbols
A1R@ment
Weak
8 characters containing:
Lower case
Upper case
Numbers
Symbols
A1R@mentS
Medium
9 characters containing:
Lower case
Upper case
Numbers
Symbols
A1R@mentSympatyczny
Very Strong
19 characters containing:
Lower case
Upper case
Numbers
Symbols
Time to crack your password:
313 trillion years
Review: Fantastic, using that password makes you as secure as Fort Knox.