

# Kompresja i szyfrowanie danych

## 1. Na czym polega kompresja danych?

### 2. Rodzaje kompresji

- 2.1. Algorytmy kompresji stratnej
- 2.2. Algorytmy kompresji bezstratnej

### 3. Szyfrowanie

- 3.1. Szyfry podstawieniowe
- 3.2. Szyfry przestawieniowe
- 3.3. Szyfry z kluczem
- 3.4. Szyfry z kluczem jawnym
- 3.5. Jednokierunkowa funkcja skrótu
- 3.6. Wykorzystanie algorytmów szyfrowania w podpisie elektronicznym

### Warto powtórzyć

1. W jakim celu wykonuje się kompresję plików i folderów?
2. Jakich programów używamy do kompresji danych? Podaj przykłady.
3. Na czym polega szyfrowanie danych (temat C3, *Informatyka podstawowa*)?
4. Co to jest podpis elektroniczny (temat A5, *Informatyka podstawowa*)?

## 1. Na czym polega kompresja danych?

Kompresja (z łac. *compressio* – „ściśnięcie”) w różnych dziedzinach nauki oznacza zmniejszenie objętości (np. w fizyce – objętości cieczy, gazów). W informatyce kompresja odnosi się do zmniejszania objętości (wielkości) danych. Nie każde zmniejszenie objętości danych jest jednak kompresją.

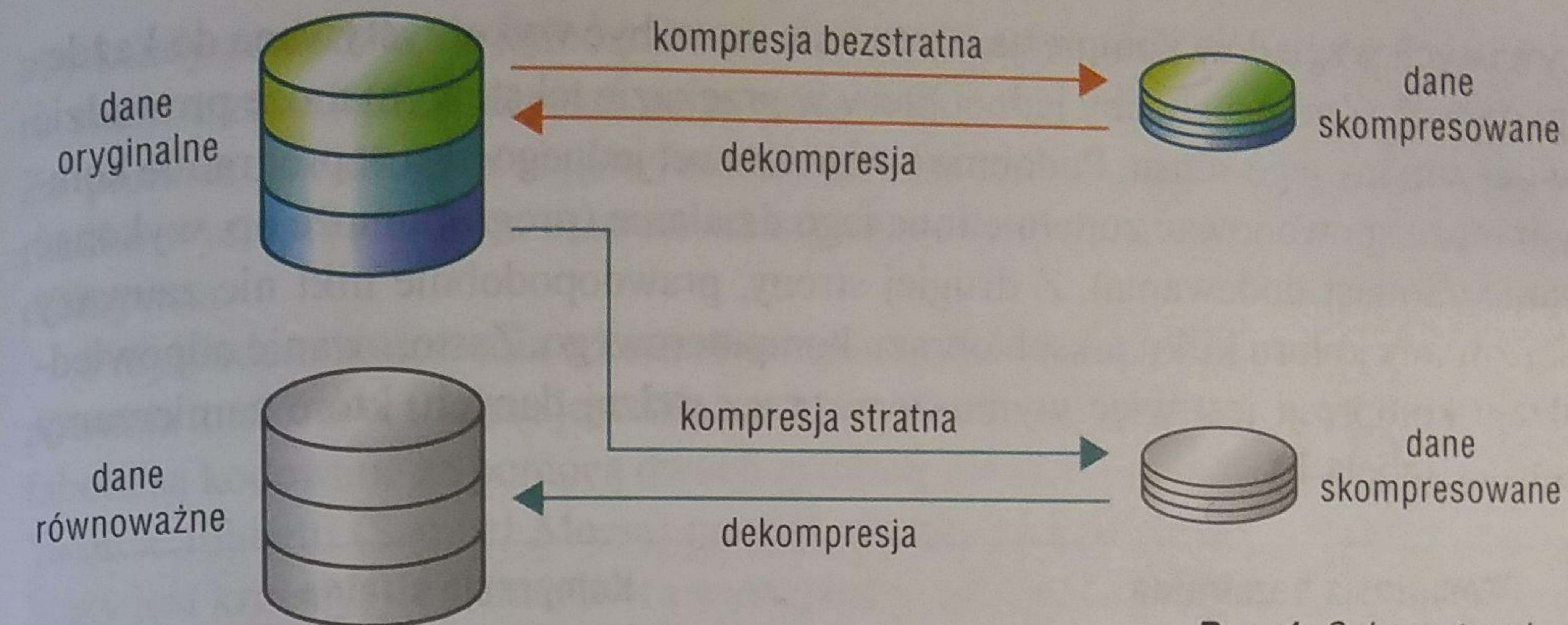


**Kompresja** to taki proces zmniejszania objętości danych, który umożliwia odtworzenie pierwotnych danych. Proces odtwarzania pierwotnych danych nazywamy **dekompresją**.

Dane odtworzone nie muszą być dokładnie takie same jak oryginalne – pewne metody kompresji powodują usunięcie niektórych fragmentów danych.

Dla każdego zbioru danych istnieje minimalna objętość, do jakiej dane te mogą zostać skompresowane. Oznacza to, że danych raz skompresowanych nie da się najczęściej ponownie skompresować.

Zastosowania kompresji danych są bardzo różnorodne. Bez kompresji danych nie istniałyby standardy JPEG, DVD, Blu-ray czy MP3. Kompresja pozwala także na efektywniejsze wykorzystywanie łączów telekomunikacyjnych (jest np. stosowana w modemach).



Rys. 1. Schemat wykonywania kompresji i dekompresji danych

**Współczynnik kompresji** obliczamy, dzieląc objętość danych skompresowanych przez objętość danych nieskompresowanych.

Korzystamy z następujących wzorów:

$$R_c = \frac{V_k}{V_{nk}} \cdot 100\% \quad [1]$$

lub

$$R'_c = \left(1 - \frac{V_k}{V_{nk}}\right) \cdot 100\%, \quad [2]$$

gdzie:

$R_c, R'_c$  – współczynniki kompresji,

$V_k$  – objętość danych skompresowanych (w bajtach),

$V_{nk}$  – objętość danych nieskompresowanych (w bajtach).

Wzór [1] pozwala nam obliczyć, jak bardzo dane zmniejszyły się w stosunku do oryginalnych, a wzór [2] – ile miejsca zaoszczędziliśmy w stosunku do oryginału.

Ze współczynnikiem kompresji wiąże się pojęcie **bitrate**. Stosuje się je w przypadku strumienia danych, a więc ciągu danych przesyłanych w czasie rzeczywistym (np. filmów, muzyki). W takim przypadku musimy dysponować urządzeniem zdolnym do przesłania w czasie jednej sekundy określonej liczby bitów. Przesłanie danych będzie możliwe tylko wtedy, gdy zostaną one odpowiednio zmniejszone. Bitrate podajemy najczęściej w bitach na sekundę, np. 128 Kb/s oznacza, że dane zostały zmniejszone tak, by jedna sekunda filmu lub dźwięku zajmowała nie więcej niż 128 kilobitów.

## 2. Rodzaje kompresji



Wyróżniamy dwa podstawowe rodzaje kompresji: **kompresję bezstratną** (ang. *lossless compression*) i **kompresję stratną** (ang. *lossy compression*).

W przypadku kompresji bezstratnej dane odtworzone są identyczne (bit po bieżącym) z danymi pierwotnymi. Natomiast w przypadku kompresji stratnej dane odtworzone są podobne do danych pierwotnych i na ogół różnią się od nich w sposób trudny do wychwycenia.

Z oczywistych względów kompresja stratna nie może być wykorzystywana do każdego rodzaju danych. Zmiana choćby jednej litery w przekazie tekstowym może prowadzić do całkowitej zmiany jego sensu. Podobnie zmiana nawet jednego bitu w programie komputerowym może spowodować zupełnie inne jego działanie (program może np. wykonać odejmowanie zamiast dodawania). Z drugiej strony, prawdopodobnie nikt nie zauważycie niewielkiej zmiany koloru kilku pikseli obrazu komputerowego. Zastosowanie odpowiedniego rodzaju kompresji jest więc wymuszone przez rodzaj danych, które zamierzamy skompresować (tabela 1.).

Kompresja bezstratna	Kompresja stratna
<ul style="list-style-type: none"><li>• teksty</li><li>• programy komputerowe</li><li>• bazy danych</li><li>• pliki z innymi danymi (pliki arkusza kalkulacyjnego, pliki konfiguracyjne, dane przesyłane w sieciach komputerowych itp.)</li><li>• niektóre rodzaje grafiki (np. formaty GIF i TIFF)</li></ul>	<ul style="list-style-type: none"><li>• dźwięki (kompresja GSM wykorzystywana w telefonii komórkowej)</li><li>• muzyka (format MP3)</li><li>• obrazy (format JPEG)</li><li>• filmy (format MPEG)</li></ul>

**Tabela 1.** Przykłady stosowania odpowiedniego rodzaju kompresji do rodzaju danych

## 2.1. Algorytmy kompresji stratnej

Algorytmy kompresji stratnej bazują na niedoskonałościach ludzkich zmysłów. Nie dostrzegamy np. niewielkich zmian barw lub drobnych różnic w fakturze powierzchni na obrazie. Jeżeli w danym momencie brzmi jednocześnie wiele dźwięków, słyszemy tylko niektóre z nich, itd. Algorytmy kompresji stratnej bazują na tym spostrzeżeniu i po prostu usuwają informacje o tych elementach, których nie jesteśmy w stanie dostrzec lub usłyszeć. Ponieważ takich informacji jest zaskakująco wiele, kompresja stratna jest bardzo efektywna.

Objętość pliku z obrazem lub dźwiękiem można zmniejszyć nawet dziesięciokrotnie bezauważalnego dla obserwatora lub słuchacza pogorszenia jakości; dopiero przy dalszym zmniejszaniu objętości różnice zaczynają być odczuwalne. Do utraty jakości może też prowadzić wielokrotne powtarzanie cyklu kompresji i dekompresji, np. odczytywanie i zapisywanie obrazka w formacie JPEG.

## 2.2. Algorytmy kompresji bezstratnej

Algorytmy kompresji bezstratnej można podzielić na dwie grupy: **statystyczne** i **słownikowe**.

**Algorytmy statystyczne** operują na pojedynczych blokach danych (np. znakach tekstu lub fragmentach obrazu o określonej wielkości). Bazują one na takich faktach, jak np. ten, że znaki w tekście występują z różną częstością. W przypadku tekstu zapisanego w języku polskim najczęściej występują samogłoski „a”, „e”, „i”, „o”, litery takie, jak „ż” czy „ń” pojawiają się rzadko, natomiast znaki takie, jak „&” czy „#” występują zupełnie sporadycznie. Do zapisania każdego znaku przy użyciu kodu ASCII potrzebujemy dokładnie 8 bitów. Jeżeli jednak do zapisu najczęściej używanych znaków wykorzystamy ciągi bitów o mniejszej długości (np. 2 lub 3 bity), zaoszczędzimy dużo miejsca. Oczywiście,

aby każdy znak dało się jednoznacznie zidentyfikować, niektóre znaki trzeba będzie zapisać za pomocą więcej niż 8 bitów. Za pomocą tak długich ciągów można jednak zapisać znaki rzadko występujące, więc ostatecznie tego typu kompresja jest efektywna.

Przykładem kodowania tekstu opartego na częstotliwości występowania znaków w tekście jest alfabet Morse'a. Znaki tego alfabetu są kodowane za pomocą dwóch symboli: kropki i kreski. Twórca alfabetu (Samuel Morse) przyjął zasadę, że kod danej litery jest krótszy, jeśli dana litera występuje w tekście częściej. Na przykład: kod litery B to: – • • •, a kod litery A, która występuje częściej niż B, to: • –. W alfabetie Morse'a nie są rozróżniane małe i wielkie litery.

Najbardziej znanym algorytmem kompresji statystycznej jest tzw. kodowanie Huffmana, w którym używa się wyłącznie zer i jedynek. Kody są tworzone tak, aby żaden nie był początkowym fragmentem innego (co pozwala na ich rozróżnianie) oraz aby średnia długość kodu danego znaku była możliwie najmniejsza.

Litera	Kod Morse'a										
a	- -	f	... - .	k	- - -	p	. - - .	u	.. -		
b	- - - -	g	- - - .	l	- - - -	q	- - - - -	v	- - - -		
c	- - - .	h	... - -	m	- - -	r	- - .	w	.. - -		
d	- - -	i	.. -	n	- - .	s	... - -	x	- - - -		
e	.	j	- - - - .	o	- - - -	t	-	y	- - - -		
								z	- - - -		

**Tabela 2.** Alfabet Morse'a – fragment z literami alfabetu łacińskiego



## Przykład 1. Alfabet Morse'a

by odczytać tekst zakodowany za pomocą alfabetu Morse'a, wystarczy zastąpić kolejne kody odpowiednimi literami.

W alfabetie Morse'a zapis: • • / - • / • • - • / - - / • - • / - - / • - / - / - • - - / - • - / •  
znacza INFORMATYKA

**waga:** Znakiem „/” oddzielamy kody liter, a znakiem „//” – wyrazy



## wiczenie 1.

Być może chcesz znać, jakiego tekstu zapisano w telegrafie nadanym w 1844 roku? Odkoduj napis:

- - / ● ● ● ● / ● - / - / ● ● ● ● / ● - / - / ● ● ● ● / / - - ● / - - - / - ● ● / / ● - - / ● - ● / - - - / ● ● - /  
- ● / ● ● ● ● / -



## wiczenie 2.

akoduj w alfabetie Morse'a swoje imię.

Algorytmy słownikowe operują na ciągach bitów o zmiennej długości (w szczególności na ciągach znaków). Patrząc np. na tekst zapisany w języku polskim, łatwo zauważyc, że pewne ciągi znaków często się powtarzają (np. w tym akapicie słowo „na” ze spacją po-awiło się już trzy razy). Teoretycznie można więc utworzyć słownik zawierający wszystkie możliwe ciągi znaków, a następnie zamiast znaków tekstu posługiwać się numerami

słów w słowniku. W praktyce tekst jest przeglądany od początku do końca i jeżeli algorytm natrafi na ciąg znaków, który pojawił się już wcześniej, zamiast całego tekstu zapisywana jest jedynie informacja o miejscu, w którym wcześniej określony ciąg znaków się pojawił.



## Przykład 2. Stosowanie algorytmu słownikowego

Zdanie: *The rain in Spain falls mainly on the plain.* (44 bajty)

po skompresowaniu może mieć postać:

*The rain <3,3>Sp<9,4>falls m<11,3>ly on t<34,3>pl<15,3>.* (33 bajty).

Zapis  $\langle x, y \rangle$  (tzw. token) oznacza, że dana kombinacja znaków pojawiła się  $x$  znaków wcześniej i miała długość  $y$  znaków. W powyższym przykładzie do zapisania  $x$  wystarczy 6 bitów, natomiast do zapisania  $y$  potrzebne są 2 bity (maksymalne  $y$  to 4 bity, jednak nie ma sensu zaznaczać wystąpienia ciągu o długości 0 znaków). Do zapisania jednego tokenu wystarczy więc 1 bajt.

Współczynnik kompresji dla powyższego przykładu wynosi  $33/44 \cdot 100\% = 75\%$ .

Najbardziej znanym algorymem kompresji słownikowej, na którym bazują algorytmy wykorzystywane w programach takich, jak WinZIP, WinRAR, jest algorytm LZW (od nazwisk autorów: Lempel-Ziv-Welch).

## Ćwiczenie 3.

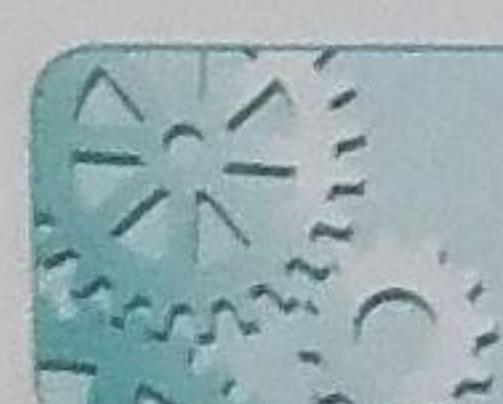


Za pomocą metody słownikowej dokonaj kompresji zdania: „Moją pasją są kompresje i ich komputerowe wersje”. Oblicz współczynnik kompresji.

## 3. Szyfrowanie

### Szyfr

to algorytm  
szyfrowania  
i deszyfrowania.



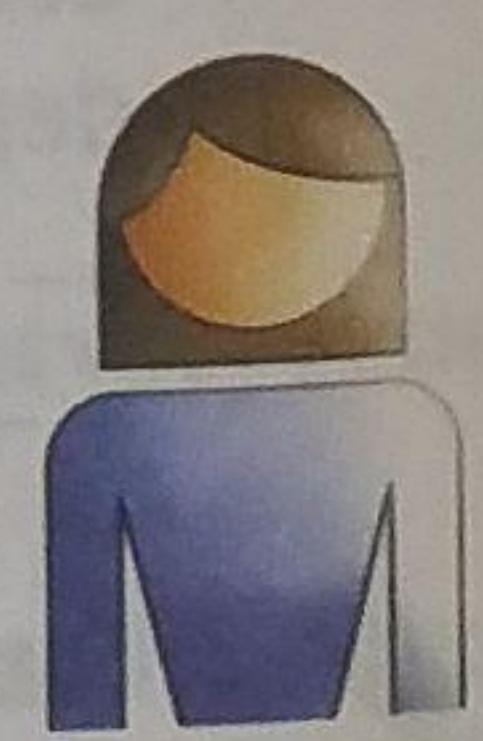
**Szyfrowanie** to proces przetwarzania wiadomości, pozwalający na ukrycie jej tekstu, natomiast **deszyfrowanie** to proces odtwarzania treści szyfrogramu.



tekst  
jawny

.....  
szyfrowanie

oryginalny  
tekst  
jawny



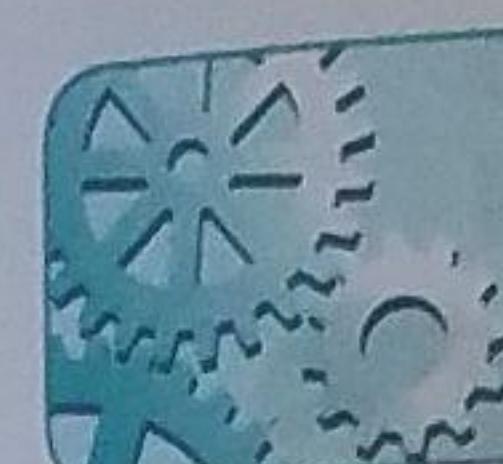
Odbiorca

Rys. 2. Schemat procesu szyfrowania i deszyfrowania

Nauka zajmująca się zabezpieczaniem wiadomości to **kryptografia**, natomiast **kryptanaliza** to nauka o łamaniu tychże zabezpieczeń. Razem tworzą one **kryptologię** – osobną dziedzinę matematyki. Szyfrowanie i deszyfrowanie wykonuje się za pomocą odpowiednich funkcji matematycznych, często przy użyciu komputera.

Szyfrowanie jest obecnie powszechnie wykorzystywane. Między innymi istota handlu elektronicznego opiera się na zapewnieniu poufności przesyłanych danych. Podobnie przed dostępem niepowołanych osób chronimy inne informacje: korespondencję osobistą, dane firmowe, wyniki badań itp.

## 3.1. Szyfry podstawieniowe



**Szyfr podstawieniowy** to taki, w którym każdy znak tekstu jawnego jest zastępowany innym znakiem.

Najbardziej znanym przykładem tekstu podstawieniowego jest **szyfr Cezara**, którego autorstwo przypisuje się Juliuszowi Cesarowi. Szyfr ten polega na zastąpieniu każdej litery tekstu jawnego znakiem stojącym w alfabetie o trzy pozycje w prawo względem znaku źródłowego. I tak „A” zostaje zastąpione przez „D”, „B” przez „E”, ... „W” przez „Z”, „X” przez „A”, „Y” przez „B” i „Z” przez „C”:

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

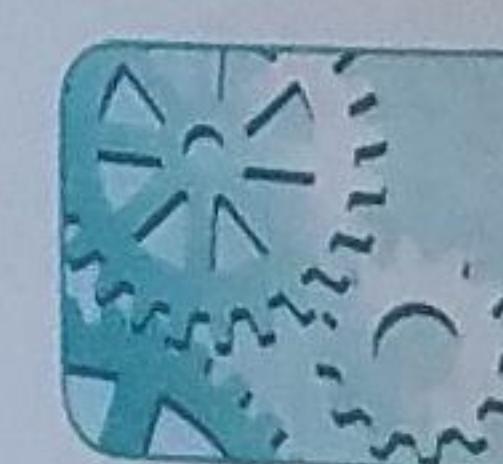
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W



## Ćwiczenie 4.

Korzystając z szyfru Cezara, zaszyfruj zdanie: „Szyfr Cezara to najbardziej znany szyfr podstawieniowy”.

## 3.2. Szyfry przestawieniowe



**Szyfr przestawieniowy** to taki, w którym w tekście zaszyfrowanym pojawiają się wszystkie znaki tekstu jawnego, ale w innej kolejności.

Szyfr przestawieniowy najprościej zrealizować, zapisując tekst jawnego wierszami o ustalonej długości, a czytając go kolumnami.



## Przykład 3. Stosowanie szyfru przestawieniowego

Tekst jawnny: TO BE OR NOT TO BE THAT IS THE QUESTION

T O B E O R N O T T  
O B E T H A T I S T  
H E Q U E S T I O N

Czytając kolumnami, otrzymujemy:

T O H O B E B E Q E T U O H E R A S N T T O I I T S O T T N

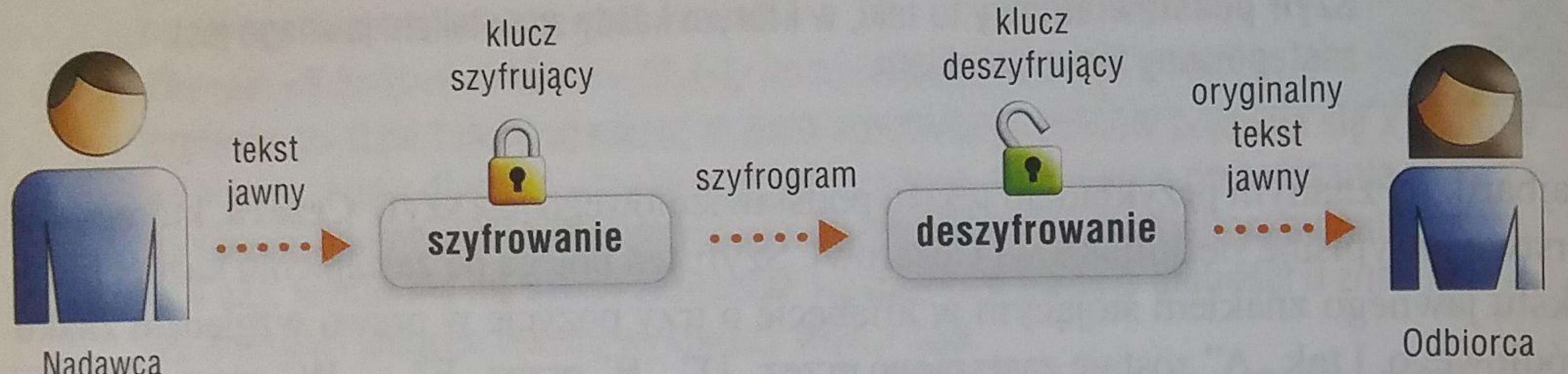
To be or not to be –  
that is the question  
– W. Szekspir, Hamlet,  
akt III, scena 1.

## 3.3. Szyfry z kluczem

Wadą powyższych systemów szyfrowania jest to, że ich bezpieczeństwo jest uzależnione od tajności użytego algorytmu szyfrowania. Poznawszy algorytm, można odszyfrować dowolną zaszyfrowaną nim wiadomość. Wady tej są pozbawione **algorytmy z kluczem**.

W tego typu algorytmach do odczytania wiadomości potrzebna jest, oprócz znajomości sposobu szyfrowania, znajomość klucza. Znając sposób szyfrowania, lecz nie znając klucza, nie można odczytać wiadomości. Dodatkowo dla każdej wiadomości klucz może być inny, więc odczytanie jednej wiadomości nie oznacza, że będzie możliwa odczytanie również inne. Wszystkie nowoczesne sposoby szyfrowania wykorzystują klucze.

W zależności od algorytmu klucz szyfrujący może być taki sam jak klucz deszyfrujący lub też mogą one być różne.



Rys. 3. Schemat szyfrowania z kluczem

Przykładem prostego szyfru z kluczem jest **szyfr Vigenère'a**. W szyfrze tym każdy znak tekstu jawnego jest szyfrowany przy użyciu szyfra podstawieniowego, jednak dla każdego znaku przesunięcie w alfabetie jest inne i zależy od wartości odpowiedniego znaku klucza.



#### Przykład 4. Stosowanie szyfru Vigenère'a

Każdemu znakowi alfabetu (dla uproszczenia posłużymy się alfabetem łacińskim) przypiszemy okreoloną wartość liczbową: A – 1, B – 2, ... Z – 26.

Szyfrując słowo „TEKST” kluczem „KLUCZ”:

literę „T” zastąpimy literą leżącą w alfabetie o 11 (wartość „K”) miejsc dalej, literę „E” – literą leżącą o 12 (wartość „L”) miejsc dalej itd.

T <sub>20</sub>	E <sub>5</sub>	K <sub>11</sub>	S <sub>19</sub>	T <sub>20</sub>
K <sub>11</sub>	L <sub>12</sub>	U <sub>21</sub>	C <sub>3</sub>	Z <sub>26</sub>
E <sub>20+11=31=26+5</sub>	Q <sub>5+12=17</sub>	F <sub>11+21=32=26+6</sub>	V <sub>19+3=22</sub>	T <sub>20+26=46=26+20</sub>

Można postużyć się tabelą 3.

		Znak tekstu jawnego																									
		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
Znak klucza	K	L	M	N	O	P	Q	R	S	T	U	V	W	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	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	Z	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

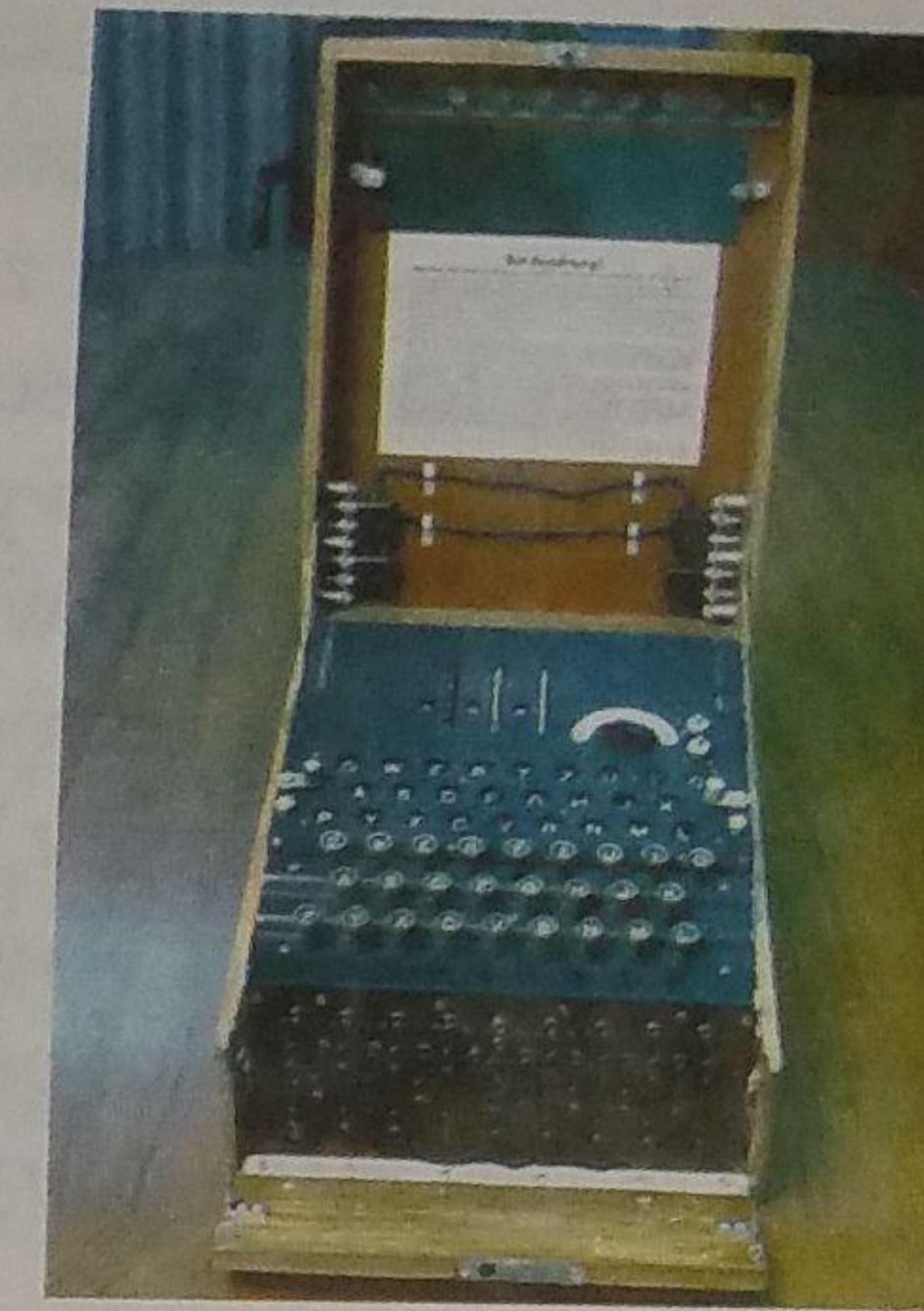
Tabela 3. Stosowanie szyfru Vigenère'a

Szyfrując w ten sposób, sumuje się wartości liczbowe każdego znaku tekstu jawnego i odpowiadającego mu znaku klucza. W ten sposób otrzymuje się wartość liczbową znaku szyfrogramu; jeżeli otrzymana wartość jest większa od 26, należy od niej

odjąć 26. Jeżeli długość klucza jest mniejsza od długości tekstu jawnego, klucz należy powielić odpowiednią ilość razy.

Deszyfrowanie polega na odjęciu od wartości liczbowej odpowiedniego znaku szyfrogramu wartości liczbowej odpowiedniego znaku klucza; jeżeli otrzymana wartość jest ujemna lub zerowa, należy dodać do niej 26. W ten sposób otrzymujemy wartość liczbową znaku tekstu oryginalnego.

Działający na podobnej zasadzie, choć oczywiście o wiele bardziej skomplikowany sposób szyfrowania, został wykorzystany przez Niemców przed II wojną światową w maszynie szyfrującej Enigma. Łamanie szyfrów Enigmy wymagało urządzeń obliczeniowych o dużej wydajności i było jednym z powodów budowy pierwszych komputerów. Szyfr Enigmy złamali w latach 30. XX w. polscy matematycy: Marian Rejewski, Jerzy Różycki i Henryk Zygalski.



Rys. 4. Niemiecka maszyna szyfrująca Enigma

#### Ćwiczenie 5.

Zaszyfruj nazwisko „KOWALSKI” kluczem „KLUCZ” za pomocą sposobu omówionego w przykładzie 4.

### 3.4. Szyfry z kluczem jawnym

Zasadniczą słabością algorytmów z kluczem jest to, że nadawca musi przekazać klucz odbiorcy. Wprawdzie mogą się oni wcześniej umówić, jakie klucze będą stosować, jednak i tak niezbędny jest kontakt nadawcy i odbiorcy. Z tego powodu proces komunikacji jest zawodny i nie zapewnia całkowitego bezpieczeństwa przekazu.

Wady tej pozbawione są algorytm z **kluczem jawnym**. W algorytmach tych klucz szyfrujący jest inny niż klucz deszyfrujący. Klucz szyfrujący (zwany **kluczem jawnym** lub **kluczem publicznym**) można udostępnić każdemu, w celu zaszyfrowania wiadomości przeznaczonych dla odbiorcy. Jednak pasującym do klucza jawnego kluczem deszyfrującym (tzw. **kluczem tajnym** lub **kluczem prywatnym**) dysponuje tylko odbiorca wiadomości. Dzięki temu tylko on może odczytać zaszyfrowaną wiadomość.

Najbardziej znanym algorymem z kluczem jawnym jest algorytm **RSA** (od nazwisk twórców: Rivest-Shamir-Adelman). Algorytm ten jest powszechnie stosowany w wielu produktach, takich jak pakiet szyfrowania PGP (ang. *Pretty Good Privacy*) czy protokół bezpiecznych połączeń SSL, wykorzystywany przez serwery i przeglądarki internetowe.

### 3.5. Jednokierunkowa funkcja skrótu

Jednokierunkowa funkcja skrótu przekształca dowolnie długi ciąg danych w krótki ciąg danych określonej długości.

Funkcja ta powinna być tak dobrana, by zmiana nawet jednego bitu w ciągu źródłowym powodowała zmianę także w ciągu wynikowym. Dzięki istnieniu takiej funkcji łatwe jest

np. sprawdzenie, czy podczas transmisji dane nie uległy celowemu bądź przypadkowemu zniekształceniu. Wystarczy bowiem obliczyć wartość funkcji skrótu dla przesłanego dokumentu i porównać ją z opublikowaną (np. w Internecie) wartością dla dokumentu oryginalnego. Jeżeli wartości się różnią – dokument został zmieniony.

Funkcje skrótu są też wykorzystywane w sieciowych systemach operacyjnych do zarządzania hasłami użytkownika. Hasło takie nie jest pamiętane nigdzie w pełnej postaci, a jedynie w postaci skróconej. Przy logowaniu hasło wpisane przez użytkownika jest skracane i porównywane z pamiętanym w systemie skrótem. Z postaci skróconej nie da się natomiast odzyskać oryginalnego hasła użytkownika, nie ma więc niebezpieczeństwa, że pozna go w ten sposób osoba nieupoważniona.

Najbardziej znane algorytmy tego typu to CRC, MD5 i SHA.

### 3.6. Wykorzystanie algorytmów szyfrowania w podpisie elektronicznym

Technologie szyfrowania umożliwiają także stosowanie **podpisu elektronicznego**.

Praktyczne formy podpisów elektronicznych stały się dostępne dzięki rozwojowi kriptografii z kluczem publicznym i polegają na dołączeniu do dokumentu jego skrótu (wygenerowanego za pomocą jednokierunkowej funkcji skrótu), zaszyfrowanego kluczem prywatnym strony podpisującej. Niezbędne jest zatem wygenerowanie pary kluczy: prywatnego i publicznego. Wygenerowany klucz publiczny przedstawia się następnie do **certyfikacji** właściwemu urzędowi. Po dostarczeniu wszystkich niezbędnych danych, potwierdzających naszą tożsamość, otrzymujemy certyfikat i możemy korzystać z podpisu elektronicznego.

Certyfikat umożliwia jednoznaczne związanie klucza publicznego z danymi posiadacza certyfikatu.

Klucz publiczny musi być udostępniony (jak nazwa wskazuje – upubliczniony). Takie klucze są przechowywane na specjalnych serwerach, tzw. serwerach kluczy.

Klucz prywatny przechowuje się zwykle na nośniku elektronicznym w postaci pliku, ale dodatkowo jest on zabezpieczony hasłem znanym tylko właścielowi klucza. Klucz prywatny trzeba szczególnie chronić. O wygenerowanie klucza prywatnego i publicznego oraz otrzymanie certyfikatu można się zwrócić do urzędu certyfikacyjnego.

Wiadomość zaszyfrowana kluczem prywatnym może być deszyfrowana tylko właściwym (z tej samej pary) kluczem publicznym. Adresat tak deszyfrowanej wiadomości ma pewność, że pochodzi ona od nadawcy posiadającego klucz prywatny, zawarty w certyfikacie.

W sytuacji odwrotnej – wiadomość zaszyfrowana kluczem publicznym może być deszyfrowana tylko przez osobę posiadającą klucz prywatny. Taką wiadomość może przesłać do nas każdy, kto zna nasz klucz publiczny. Nadawca ma pewność, że dotrze ona tylko do osoby posiadającej klucz prywatny (nie trafi w niepowołane ręce).

- Aby zabezpieczyć informacje, można posługiwać się różnymi algorytmami szyfrującymi, m.in. algorytmem podstawieniowym, przestawieniowym, szyfrem z kluczem.

### Pytania, problemy

1. Podaj przykład kompresji statystycznej niezwiązanej z komputerami.
2. Na czym polega kompresja danych?
3. W jakim celu zwykle kompresujemy dane? Podaj kilka przykładów.
4. Wskaż na przykładzie różnice między kompresją strażną i bezstrażną.
5. W jaki sposób działają techniki szyfrowania wiadomości? Podaj kilka przykładów.
6. Omów główną ideę algorytmów statystycznego i słownikowego.
7. Wyjaśnij na wymyślonym przez siebie przykładzie, na czym polega szyfrowanie podstawieniowe i przestawieniowe.
8. W jaki sposób wykorzystuje się szyfrowanie w podpisie elektronicznym?



### Zadania

1. Odszyfruj tekst: „VCBIU FHCDUD MHVW SURVWB”.
2. Korzystając z metod szyfrowania omówionych w tym temacie, zaszyfruj na dwa sposoby swoje dane osobowe (imię, nazwisko, miejsce zamieszkania).
3. Przedstaw szyfrowanie Vigenère'a w arkuszu kalkulacyjnym.  
**Wskazówka:** Każdą literę tekstu i klucza umieść w osobnej komórce, wykorzystaj funkcje KOD, ZNAK i JEŻELI.
4. Napisz w wybranym języku programowania program umożliwiający zaszyfrowanie bądź odszyfrowanie szyfrem Cezara tekstu wprowadzonego przez użytkownika.

### Dla zainteresowanych

5. Napisz w wybranym języku programowania program obliczający i wypisujący częstotliwość występowania liter języka polskiego. Zmierz tę częstotliwość na podstawie tekstu *Pana Tadeusza* (znajdziesz go w Internecie).
6. Napisz w wybranym języku programowania program szyfrujący i deszyfrujący szyfrem przestawieniowym kolumnowym tekst wprowadzony przez użytkownika. Szerokość kolumny powinna być definiowana przez użytkownika.
7. **Steganografia** to nauka o ukrywaniu informacji wewnętrz innej informacji. Zapoznaj się z tym zagadnieniem, korzystając z dodatkowej literatury i źródeł internetowych.
8. Korzystając z Internetu i dodatkowej literatury, odszukaj informacje na temat Enigmy.
9. Korzystając z Internetu i dodatkowej literatury, dowiedz się więcej na temat:
  - a. kodu Huffmana,
  - b. algorytmu z kluczem jawnym RSA.



### Warto zapamiętać

- Kompresję danych przeprowadzamy w celu minimalizacji objętości danych, np. przygotowując plik załącznika w poczcie elektronicznej.
- Wyróżniamy dwa rodzaje kompresji: bezstrażną (odtworzone dane są identyczne z pierwotnymi) oraz strażną (odtworzone dane są podobne do pierwotnych).