



## Łańcuchy znaków

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Gra edukacyjna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



## łańcuchy znaków

Źródło: Danielle MacInnes, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Co łączy ze sobą czynności: zmiany wielkich liter w tekście na małe, wyszukiwania słowa czy liczenia wykorzystanych znaków? Wszystkie są operacjami wykonywanymi na łańcuchach znaków. Wykorzystywanie w praktyce wiedzy na temat łańcucha znaków (tablicy znaków) oraz umiejętności odpowiedniego operowania na znakach jest codziennością programisty.

W tym e-materiale poszerzysz swoją wiedzę dotyczącą reprezentacji łańcuchów znaków oraz poznasz funkcje, które umożliwiają operacje na znakach.

Jak wyglądają operacje na łańcuchach znaków w poszczególnych językach programowania dowiesz się w e-materiałach:

- [Łańcuchy znaków w języku C++](#),
- [Łańcuchy znaków w języku Python](#),
- [Łańcuchy znaków w języku Java](#).

Więcej zadań? Przejdź do e-materiału [Łańcuchy znaków – zadania maturalne](#).

**Twoje cele**

- Scharakteryzujesz jeden z najbardziej popularnych systemów kodowania znaków – tabelę ASCII.
- Wyjaśnisz, czym jest rzutowanie zmiennych oraz w jaki sposób rzutować zmienne na konkretne typy w wybranych językach programowania.
- Przeanalizujesz funkcje i operacje przeprowadzane na łańcuchach znaków.

# Przeczytaj

---

## Łańcuch znaków

Łańcuch znaków to ciąg znaków o ustalonej wcześniej lub dowolnej długości, w którym znaki to kolejno zapisane bajty w pamięci.

Przykłady łańcuchów znaków:

- „Krokodyl”,
- „Informatyka jest super”,
- „1234453ABC”,
- „#\*%\$. /#”.

## Tablica ASCII

Przyjrzyjmy się tabeli ASCII dla zakresu wartości od 32 do 127.

Wartość	Symbol	Wartość	Symbol	Wartość	Symbol
32	Spacja	64	@	96	`
33	!	65	A	97	a
34	„	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o

Wartość	Symbol	Wartość	Symbol	Wartość	Symbol
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_	127	Delete

Wartości z zakresu 0-31 to w większości (ale nie wszystkie) klawisze kontrolne – Esc, tabulator, Backspace itd. Natomiast zakres wartości 128-255 obejmuje znaki charakterystyczne dla danego języka – w języku polskim są to m.in. „ł”, „ć”, „ą”, „ż”.

### Ciekawostka

Przeznaczenie pierwszych 32 kodów (0-31) ASCII znakom sterującym funkcjonuje już od czasów, gdy komputery były jedynie terminalami („odbiornikami” treści z serwera). Wówczas opisywane kody umożliwiały sterowanie urządzeniem (terminalem), np. służyły do manipulowania ekranem lub papierem w drukarce.

Kody ASCII w systemie binarnym reprezentowane są na 8 bitach, np. literze „H” odpowiadać będzie kod „0100 1000”. Analogicznie cyfry w kodzie ASCII mają konkretną reprezentację binarną, odmienną niż w przypadku rozpatrywania ich jako zmiennych liczbowych.

### Przykład 1

Cyfrę 9 możemy zaprezentować w systemie binarnym jako 1001, natomiast wartość 9 w kodzie ASCII (binarnie 0000 1001) przypisana jest tabulatorowi. Znakowi „9”

odpowiada inna wartość – 57 (binarnie 0011 1001). Możemy więc zauważyć, że reprezentacje binarne cyfry 9 i kodu ASCII dla znaku „9” będą się od siebie różnić.

## Rzutowanie zmiennych

Rzutowanie zmiennych znane jest także pod pojęciem **konwersja typu**. Jest to zmiana typu zmiennej. Czynność ta jest bardzo często wykorzystywana podczas programowania.

Założmy, że mamy zmienną typu znakowego.

```
1 zmienna = 'C'
```

W `zmienna` przechowujemy znak `'C'`. Zastanów się, co zostanie wyświetlone, po wykonaniu następującej operacji:

```
1 zmienna = 'C'  
2  
3 wyświetl(rzutujCałk(zmienna))
```

### Ważne!

- funkcja `rzutujCałk()` wykonuje poprawne rzutowanie do typu liczbowego całkowitego,
- funkcja `wyświetl()` powoduje wyświetlenie [parametru](#).

Zostanie wyświetlona liczba 67, która jest odpowiednikiem dziesiętnym znaku `'C'` w tablicy ASCII.

### Praca domowa

Zastanów się, co zostanie wyświetlone, jeżeli wykonamy przedstawiony za pomocą pseudokodu algorytm. Poznaj definicje metod:

- `rzutujCałk()` wykonuje poprawne rzutowanie do typu liczbowego całkowitego,
- `bin()` konwertuje liczbę typu liczbowego całkowitego na liczbę zapisaną binarnie,
- `wyświetl()` powoduje wyświetlenie parametru.

```
1 zmienna = '#'  
2  
3 wyświetl(bin(rzutujCałk(zmienna)))
```

Wskazówka: tablica ASCII.

### **Ważne!**

Opisane powyżej rzutowanie odnosi się do języków programowania obsługujących zmienne typu `char`, np. języka C++. Podobny efekt w języku Python otrzymać można, wykorzystując funkcję `ord()`. Jej działanie nie jest jednak rzutowaniem, a zwracaniem kodu ASCII dla danego symbolu.

## **Funkcje i operacje**

Omówimy teraz typowe funkcje w językach programowania, które realizują różne rodzaje operacji na łańcuchach znaków.

### **Funkcja `length`**

Funkcja `length` jest jedną z najczęściej wykorzystywanych funkcji przy operacjach na łańcuchach znaków. Służy ona do określania długości ciągu znaków wyrażanej w liczbie znaków.

#### **Przykład 2**

Funkcja `length` dla łańcucha znaków „INFORMATYKA” zwróci wartość liczbową 11.

### **Konkatenacja**

„Sklejanie” ciągów znaków. Konkatenacja powoduje łączenie ze sobą łańcuchów znaków.

#### **Przykład 3**

Po konkatenacji łańcuchów znaków „Ala”, „ma”, „kota” powstanie łańcuch znaków „Alamakota”. Jeżeli chcielibyśmy otrzymać tekst „Ala ma kota”, wówczas należałoby dokonać konkatenacji również dodatkowych dwóch znaków spacji „ ”, umieszczonych pomiędzy ciągami „Ala” i „ma” oraz „ma” i „kota”.

#### **Ciekawostka**

W przypadku języków programowania C++, Java, Python do łączenia łańcuchów znaków używa się znaku dodawania `+`.

### **Funkcja `insert`**

Funkcja umożliwiająca wstawianie znaków i ciągów znaków w określonym miejscu. Zazwyczaj przyjmuje dwa parametry: ciąg znaków do wstawienia oraz miejsce (indeks), w którym łańcuch znaków ma zostać wstawiony.

## Funkcja erase

Funkcja usuwająca podaną część z łańcucha znaków. Występuje najczęściej w dwóch postaciach:

1. Przyjmuje tylko jeden parametr – jest to indeks znaku, który ma zostać usunięty z łańcucha znaków.
2. Przyjmuje dwa parametry, dwa indeksy, z czego pierwszy to początek wycinanego fragmentu a drugi koniec.

## Funkcja replace

Funkcja `replace` najczęściej występuje w poniższych postaciach:

1. Przyjmuje dwa parametry, którym są znaki. Funkcja zamienia wszystkie występowania znaku podanego jako pierwszy parametr na znak podany jako drugi parametr.
2. Przyjmuje dwa parametry, którymi są łańcuchy znaków. Pierwszy parametr to łańcuch znaków, który chcemy zamienić, a drugi to łańcuch znaków, na jaki chcemy zamienić.

## Funkcja find

Służy do wyszukiwania wzorca w łańcuchu znaków. Zazwyczaj funkcja zwraca pozycję (indeks) wystąpienia wzorca w łańcuchu znaków. Innym wariantem może być funkcja `find`, która jest typu logicznego i zwraca prawdę, jeżeli wzorec wystąpił w sprawdzanym łańcuchu znaków, jeśli nie – zwraca fałsz.

## Funkcja substr

Funkcja wyszukująca w ciągu znaków podciąg o podanej długości znajdujący się w ustalonej lokalizacji, a następnie go zwracająca. Jako parametry opisywana funkcja przyjmuje ciąg znaków, pozycję znaku w ciągu, od której zaczyna się podciąg oraz liczbę znaków zawartych w podciągu.

### Przykład 4

Wyszukiwanie podciągu w słowie „NAUKA”:

```
1 substr("NAUKA", 2, 3)
```

Jako wynik działania funkcji otrzymamy podciąg „AUK”.

## Funkcja transform



Aby zamienić w ciągu znaków wszystkie małe liter na wielkie, potrzebujemy w pierwszej kolejności funkcję zmieniającą wielkość wybranego znaku. Przyjmijmy, że za ten element odpowiedzialna jest funkcja `zamiana`. Działa ona w następujący sposób:

1. Dokonaj rzutowania litery do typu liczbowego całkowitego.
2. Jeżeli uzyskana liczba całkowita jest mniejsza niż 123, ale większa niż 96 (kody ASCII od 97 do 122 przeznaczone są na małe litery) odejmij od zmiennej 32 (stała różnica w tabeli ASCII pomiędzy kodami tej samej litery w wariacie wielkim i małym).
3. Dokonaj rzutowania zmiennej do typu znakowego.
4. W rezultacie otrzymujemy wielką literę.

Wiemy już, jak zmienić pojedynczy znak, jednak chcemy przeprowadzić tę operację na całym ciągu. Wykorzystamy do tego funkcję `transform`. Pozwala ona na wykonanie pożądanej operacji na zbiorze, a następnie zwrócenie rezultatów do wybranego zbioru (tego samego bądź innego).

```
1 transform(początek zbioru, koniec zbioru, początek nowego zbioru,
```

Możemy już wykorzystać obie funkcje do wykonania zadania. Dodatkowo, aby wskazać początek i koniec zbioru, wykorzystamy odpowiednio funkcje `początek` i `koniec`. Załóżmy, że zamieniamy wielkości liter w słowie „Litera” znajdującym się w zmiennej `słowo`.

```
1 słowo = "Litera"  
2  
3 transform(początek(słowo), koniec(słowo), początek(słowo), zamian
```

W wyniku powyższej operacji zmienna `słowo` będzie zawierała ciąg znaków „LITERA”.

## Apostrofy i cudzysłowy

W niektórych językach programowania, np. C++ i Java, istnieje pewna zależność dotycząca używania apostrofów i cudzysłowów.

Znaku pojedynczego apostrofu używa się przy deklaracji, przypisaniu pojedynczego znaku. W poniższym przykładzie zmienna `litera` jest zmienną znakową (`char`) i zostaje jej przypisana wartość (pojedynczy znak) przy użyciu właśnie pojedynczych apostrofów.

```
1 char litera    'c'
```

Moglibyśmy także nadać wartość zmiennej `litera` (przypisać znak „C”) w inny sposób, np. podać wartość dziesiętną tej litery. Tutaj pomocna okazuje się wcześniej omówiona tablica ASCII. Następujące linie kodu są bliźniacze:

```
1 char litera      'C'  
2  
3 char litera      67
```

W przypadku deklarowania oraz przypisywania łańcuchów znaków używane są do tego cudzysłowy. W przytoczonym przykładzie zmienna `zdanie` jest łańcuchem znaków (`string`), któremu zostaje przypisany ciąg znaków „ALA MA KOTA”.

```
1 string zdanie    "ALA MA KOTA"
```

W wypadku używania cudzysłowów wszystko jest rozpatrywane jako tekst. W związku z tym zaprezentowana linia kodu nie spowoduje, że w zmiennej `zdanie` będzie zapisana litera „C”, tylko tekst, łańcuch znaków „67”.

```
1 string zdanie    "67"
```

Warto dodać, że w języku Python rozróżnienie pomiędzy cudzysłowem i apostrofem nie istnieje, ponieważ znak i ciąg znaków są tego samego typu.

## Funkcje pobierające słowa od użytkownika

W wielu językach programowania wyróżniamy dwie główne metody pobierania łańcuchów znaków od użytkownika:

- **metoda pobierająca pojedyncze słowo, łańcuch znaków** (pobiera do pierwszej spacji, odstępu) od użytkownika i zapisuje je do zmiennej,
- **metoda pobierająca wiele łańcuchów, a także odstępów** (pobiera do znaku nowej linii).

## Słownik

**bajt**

wykorzystywana w dziedzinie informatyki jednostka pamięci złożona z 8 bitów

**parametr**

zmienna, która jest zdefiniowana przy deklaracji metody lub funkcji



# Gra edukacyjna

---

## Polecenie 1

Przeanalizuj prezentację. Dowiedz się z niej, jak stworzyć własne implementacje wybranych funkcji operujących na łańcuchach znaków.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

1

Zacznijmy od implementacji funkcji `length`, która służy do obliczania długości łańcucha znaków.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

Pierwszym krokiem będzie zadeklarowanie funkcji i zdefiniowanie, co będzie ona przyjmować jako parametr.

```
1 funkcja length(ciągZnakow)
2
3
```

Jedynym parametrem funkcji będzie ciąg znaków, dla którego funkcja będzie musiała obliczyć długość i zwrócić wartość na wyjście.

Materiał audio dostępny pod adresem:

3

<https://zpe.gov.pl/b/PpymgjpJF>

Następnym krokiem jest zadeklarowanie zmiennej wynikowej `długosc`, którą funkcja będzie zwracać. Zmienna ta, będzie przechowywać liczbę całkowitą – obliczoną długość ciągu znaków (podanego jako parametr). Ustalana jest początkowa wartość zmiennej `długosc`, równa 0.

```
1 funkcja length(ciągZnakow)
2     długosc = 0
3
4     zwróć długosc
5
```

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

Kolejnym etapem jest napisanie pseudokodu odpowiedzialnego za iterowanie po znakach zmiennej `ciągZnakow` i inkrementację zmiennej `długosc` w przypadku wykrycia znaku.

```
1 funkcja length(ciągZnakow)
2     długosc = 0
3     i = 0
4
5     jeżeli nie
znakPusty(ciągZnakow[i])
wykonuj
6         długosc++
7         i++
8
9     zwróć długosc
10
```

Została zadeklarowana zmienna `i`, która pełni rolę zmiennej sterującej pętlą, następnie w pętli

iterujemy po kolejnych znakach i w przypadku, gdy znak nie jest pusty, to zwiększamy wartość zmiennej `długość`

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

5

Przejdźmy do napisania własnej implementacji kolejnej funkcji operującej na łańcuchach znaków – funkcji `replace`.

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

Ponownie należy zadeklarować funkcję i zdefiniować, jakie parametry będzie przyjmować.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

7

W przypadku funkcji `replace` przyjmować ona będzie trzy parametry – dwa znaki oraz ciąg znaków. Pierwszym z nich będzie znak, którego wystąpienia funkcja ma zamienić na inny znak – drugi parametr.

```
1 funkcja replace(staryZnak,  
2   nowyZnak, ciągZnakow)
```

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

Następnie należy zadeklarować pętlę, w której odbywać się będzie iteracja po kolejnych znakach zmiennej `ciagZnakow`. W pętli zostanie użyta wcześniej zaimplementowana funkcja `length`. Zadeklarować również należy zmienną `i`, która będzie pełniła rolę zmiennej sterującej pętlą.

```
1 funkcja replace(staryZnak,  
2   nowyZnak, ciagZnakow)  
3   i = 0  
4   dopóki i <  
5     length(ciagZnakow) wykonuj
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PpymgjpJF>

9

Ostatnim etapem jest napisanie odpowiedniej instrukcji warunkowej, która w przypadku wykrycia znaku `staryZnak` zamieni go na `nowyZnak`.

```
1 funkcja replace(staryZnak,  
2   nowyZnak, ciagZnakow)  
3   i = 0  
4   dopóki i <  
5     length(ciagZnakow) wykonuj  
6     jeżeli  
7       ciagZnakow[i] == staryZnak  
8       wykonaj  
9         ciagZnakow[i]  
10        = nowyZnak
```

7

8

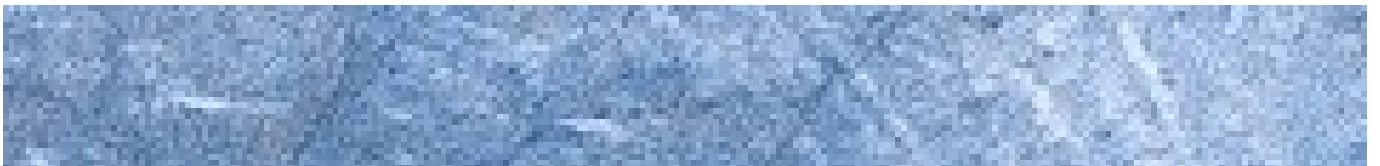
9

zwróć ciągZnakow

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

## Polecenie 2

Sprawdź swoją wiedzę, biorąc udział w grze.



Test

**Sprawdź swoją wiedzę, rozwiązując quiz**

Poziom trudności:

**łatwy**

Limit czasu:

**5 min**

Twój ostatni wynik:

-

Uruchom

## Polecenie 3

Zaproponuj trzy pytania, które mogłyby pojawić się w grze edukacyjnej.



#### Polecenie 4

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w prezentacji multimedialnej.

# Sprawdź się

---

Pokaż ćwiczenia:   

## Ćwiczenie 1



Zaznacz poprawną odpowiedź. Wskaż zakres liczb dziesiętnych przypisanych w tabeli „ASCII” znakom od „A” do „Z”.

od 97 do 127

od 32 do 58

od 65 do 90

od 0 do 25

## Ćwiczenie 2



Dokończ zdanie.

Akronim ASCII oznacza...

*American Standard Code for International Interchange.*

*American Standard Certification for Information Interchange.*

*American Super Code for Information Interchange.*

*American Standard Code for Information Interchange.*

*Autonomous Standard Code for Information Interchange.*

### Ćwiczenie 3



Wybierz poprawną odpowiedź. Mamy zmienną typu znakowego. W zmienna przechowujemy znak „f”. Zastanów się, co zostanie wyświetlone po wykonaniu następującej operacji:

```
wyświetl(rzutujCałk(zmienna))
```

Gdzie:

- - funkcja `rzutujCałk()` wykonuje poprawne rzutowanie do typu liczbowego całkowitego,
- - funkcja `wyświetl()` powoduje wyświetlenie parametru.

70

102

45

f

### Ćwiczenie 4



Uzupełnij definicję określenia bajt.

Wykorzystywana w dziedzinie informatyki   złożona z  bitów.

8

zmienna

pamięci

funkcja

jednostka

### Ćwiczenie 5



Wskaż, jak nazywa się funkcja, która operuje na łańcuchach znaków w taki sposób, że umożliwia wstawianie do nich nowych znaków oraz łańcuchów znaków.

replace

find

erase

insert

### Ćwiczenie 6



Wskaż, jak nazywa się funkcja, która operuje na łańcuchach znaków, w taki sposób, że umożliwia usuwanie pojedynczego znaku lub podanej części z łańcucha znaków.

find

erase

length

replace

## Ćwiczenie 7



Dopasuj funkcje.

Pierwszy parametr tej funkcji to łańcuch znaków, który chcemy zamienić – a drugi to łańcuch znaków, na jaki chcemy zamienić.

find

insert

replace

erase

Usunięcie jednego lub wielu elementów z listy.

Wstawianie znaków i ciągów znaków w określonym miejscu.

Szukanie występującego elementu.

## Ćwiczenie 8



Zaznacz wszystkie poprawne odpowiedzi. Wskaż, w których językach programowania do wykonania konkatenacji łańcuchów znaków używa się znaku +.

Java

Python

C++

# Dla nauczyciela

---

**Autor:** Maurycy Gast

**Przedmiot:** Informatyka

**Temat:** Łącuchy znaków

**Grupa docelowa:**

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres rozszerzony

**Podstawa programowa:**

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

5) przedstawia sposoby reprezentowania w komputerze znaków, liczb, wartości logicznych, obrazów, dźwięków, animacji;

**Kształtowane kompetencje kluczowe:**

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

**Cele operacyjne (językiem ucznia):**

- Scharakteryzujesz jeden z najbardziej popularnych systemów kodowania znaków – tabelę ASCII.
- Wyjaśnisz, czym jest rzutowanie zmiennych oraz w jaki sposób rzutować zmienne na konkretne typy w wybranych językach programowania.
- Przeanalizujesz funkcje i operacje przeprowadzane na łańcuchach znaków.

### **Strategie nauczania:**

- konstruktywizm;
- konektywizm.

### **Metody i techniki nauczania:**

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

### **Formy pracy:**

- praca indywidualna;
- praca w grupach;
- praca całego zespołu klasowego.

### **Środki dydaktyczne:**

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

### **Przebieg lekcji**

#### **Przed lekcją:**

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Łańcuchy znaków”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj” w kontekście programowania.

#### **Faza wstępna:**

1. Prowadzący wyświetla na tablicy interaktywnej zawartość sekcji „Wprowadzenie” i omawia cele do osiągnięcia w trakcie lekcji.
2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

#### **Faza realizacyjna:**

1. **Praca z tekstem.** Nauczyciel ocenia stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi chętne osoby o streszczenie najważniejszych kwestii poruszonych w sekcji „Przeczytaj”.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Gra edukacyjna”. Uczniowie indywidualnie analizują prezentację, z której dowiedzą się, jak napisać własne implementacje wybranych funkcji operujących na łańcuchach znaków.
3. **Ćwiczenie umiejętności.** Liga zadaniowa - uczniowie wykonują indywidualnie na czas ćwiczenia nr 1-4 z sekcji „Sprawdź się”, a następnie omawiają zadania na forum.
4. Ćwiczenia nr 5-8 uczniowie wykonują w grupach czteroosobowych, a następnie porównują swoje odpowiedzi z inną grupą.

#### **Faza podsumowująca:**

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).

#### **Praca domowa:**

1. Uczniowie wykonują polecenie: Przyjrzyj się temu, jak korzystasz z sieci. W jakich sytuacjach masz do czynienia z łańcuchami znaków?
2. Uczniowie wykonują polecenie 2 z sekcji „Gra edukacyjna”.

#### **Wskazówki metodyczne:**

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.