



## Zajęcia dodatkowe dla Uczniów Szkoły

### **III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.**

Tytuł zajęć

**„ Zajęcia rozwijające pasje i zainteresowanie informatyką.”**

Autor/Autorzy opracowania

**mgr Anna Jaborkhel**

Niniejszy skrypt/scenariusz powstał na potrzeby realizacji Projektu  
nr RPWP.08.01.04-30-0005/19 pn.:

*„ENIGMA – Wsparcie nauczania matematyki i informatyki  
w szkołach podstawowych i ponadpodstawowych  
Metropolii Poznań”*

Poznań 2021



## PROGRAM ZAJĘĆ

L.p.	Temat zajęć	Liczba godzin
1.	<b>Systemy liczbowe i reprezentacja danych w komputerze.</b>	5
2.	<b>Metody sortowania.</b>	3
3.	<b>Szukamy różnych podciągów.</b>	3
4.	<b>Tworzenie stron internetowych - HTML.</b>	3
5.	<b>Tworzenie stron internetowych – CSS (<i>Cascading Style Sheets</i>).</b>	4
6.	<b>Projektowanie stron z wykorzystaniem WordPress.</b>	5
Łączna liczba godzin		23

## Spis treści:

### Wybrane algorytmy programistyczne.

1. **Systemy liczbowe i reprezentacja danych w komputerze.**
  - 1.1. Systemy liczbowe stosowane w informatyce - reprezentacja liczb całkowitych, znaków i napisów w komputerze .
  - 1.2. Algorytm zamiany reprezentacji liczby dziesiętnej na dwójkową.
  - 1.3. Algorytm zamiany reprezentacji liczby dwójkowej na dziesiętną.
  - 1.4. Uogólnienie algorytmów zamiany reprezentacji liczb między systemami pozycyjnymi.
  - 1.5. Sprawdzamy czy wyraz jest palindromem.
  - 1.6. Sprawdzamy czy dwa wyrazy są anagramami.
2. **Metody sortowania.**
  - 2.1. Sortowanie bąbelkowe i przez wybieranie.
  - 2.2. Sortowanie szybkie.
  - 2.3. Sortowanie przez scalanie.
3. **Szukamy różnych podciągów.**
  - 3.1. Szukamy długości najdłuższego spójnego podciągu niemalejącego.
  - 3.2. Szukamy maksymalnej sumy podciągu spójnego.
  - 3.3. Szukamy podciągu spójnego o maksymalnej sumie.

### Tworzenie stron internetowych.

4. **HTML**
  - 4.1. Podstawowe znaczniki HTML – szkielet strony.
  - 4.2. Formatowanie treści strony za pomocą znaczników HTML.
  - 4.3. Listy, odsyłacze i grafika na stronach WWW.
5. **CSS (Cascading Style Sheets).**
  - 5.1. Podstawy CSS.
  - 5.2. Źródła stylów.
  - 5.3. Kaskadowość, dziedziczenie, specyficzność
  - 5.4. Selektory.
6. **Projektowanie stron z wykorzystaniem WordPress**
  - 6.1.1. Instalacja WordPress
  - 6.1.2. Obsługa panelu administracyjnego.
  - 6.1.3. Obsługa motywów i wtyczek.

# Wybrane algorytmy programistyczne.

## 1 Systemy liczbowe i reprezentacja danych w komputerze.

### Wiadomości wstępne.

**Systemem liczbowym** nazywamy sposób zapisywania liczb oraz zbiór reguł umożliwiających wykonywanie działań na tych liczbach.

Dla każdego systemu liczbowego istnieje zbiór znaków, za pomocą których tworzy się liczby. Znaki te zwane cyframi można zestawiać ze sobą na różne sposoby otrzymując nieskończoną liczbę kombinacji.

Najbardziej prymitywny systemem liczbowy, to system, w którym występuje tylko jeden znak.

W systemie tym kolejne liczby są tworzone przez proste powtarzanie tego znaku.

Z bardziej złożonych systemów rozróżniamy **pozycyjne** i **niepozycyjne** systemy liczbowe.

W systemach liczbowych **pozycyjnych** liczbę przedstawia się jako ciąg cyfr. Wartość jej jest zależna od położenia (pozycji) cyfry w liczbie.

Oznacza ona wielokrotność potęgi pewnej liczby uznawanej za podstawę danego systemu. Liczby zapisujemy przy pomocy cyfr od strony lewej do prawej.

System pozycyjny umożliwia zapisywanie ułamków, przy czym liczby wymierne składają się albo ze skończonej liczby znaków, albo są od pewnego miejsca okresowe.

Systemy pozycyjne posiadają pojedyncze symbole dla kilku pierwszych liczb. Cyfry te są kolejno umieszczane w ściśle określonych pozycjach i oznaczają mnożnik potęgi liczby  $n+1$  gdzie  $n$  jest najwyższą liczbą reprezentowaną pojedynczą cyfrą. W momencie, gdy dana potęga nie jest potrzebna do zapisu danej liczby, zostawia się w zapisie puste miejsce lub częściej specjalny symbol oznaczający zbiór pusty. Obecnie jest to cyfra zero.

Zbiór podstawowych cech dowolnego systemu pozycyjnego o podstawie  $p$ :

1. System pozycyjny charakteryzuje liczba zwana podstawą systemu pozycyjnego.
2. Do zapisu liczby służą cyfry.
3. Cyfr jest zawsze tyle, ile wynosi podstawa systemu:  $0, 1, 2, \dots, p-1$ .

4. Cyfry ustawiamy na kolejnych pozycjach.
5. Pozycje numerujemy od 0 poczynając od strony prawej zapisu.
6. Każda pozycja posiada swoją wagę.
7. Waga jest równa podstawie systemu podniesionej do potęgi o wartości numeru pozycji.
8. Cyfry określają ile razy waga danej pozycji uczestniczy w wartości liczby.
9. Wartość liczby obliczamy sumując iloczyny cyfr przez wagi ich pozycji.

Najbardziej znanym systemem pozycyjnym jest **system dziesiętny**, w którym za bazę przyjmuje się liczbę 10. Tym samym liczbę 23651 można przedstawić jako:  $23651 = 2 \cdot 10^4 + 3 \cdot 10^3 + 6 \cdot 10^2 + 5 \cdot 10^1 + 1 \cdot 10^0$ .

Zaletą systemów pozycyjnych jest ich klarowność, łatwość dokonywania nawet złożonych operacji arytmetycznych oraz możliwość zapisu dowolnie dużej liczby, jednak do zapisu bardzo dużych liczb jest potrzebna duża liczba cyfr.

**System dziesiętny**, którym posługujemy się na co dzień, jest znany jako system **arabski** lub **indyjsko-arabski**.

System dziesiętny został zapoczątkowany w Indiach w V w. n.e., a rozpowszechnił się w krajach arabskich dzięki matematykowi al-Chuwarizmi, który w połowie VIII w. przetłumaczył na arabski indyjską książkę o matematyce.

Dziewięć pierwszych cyfr oznaczających wartości od 1 do 9 były przedstawiane jako umowne znaki. Hindusi jako pierwsi wpadli na pomysł pisania cyfr słowami:

1	2	3	4	5	6	7	8	9
adi	dvi	tri	katur	pañca	sat	sapta	asta	nova

Muhammad Ibn Musa Al-Chuwarizmi, bibliotekarz Domu Mądrości w Bagdadzie, opisał efektywne sposoby pisemnego wykonywania obliczeń z użyciem systemu dziesiętnego, znanego wcześniej wyłącznie w Indiach. Dzięki jego pismom system ten dotarł do Europy. Od łacińskiej wersji nazwiska tego uczonego pochodzi słowo **algorytm**.



IX w.

**Inne systemy pozycyjne to np.:** system dwójkowy (binarny), trójkowy, ósemkowy, szesnastkowy.

**Systemy niepozycyjne** posiadają osobne symbole kilku liczb, a następnie posiadają kolejne symbole dla ich wielokrotności. W systemach tych liczby tworzy się przez dodawanie kolejnych symboli.

Wartość liczby jest sumą wartości poszczególnych cyfr wchodzących w jej skład. Cyfrom przypisane są stałe wartości, niezależne od ich położenia w liczbie.

**Reprezentanci systemów addytywnych to np.:** system rzymski, system alfabetyczny, system hieroglificzny;



## System rzymski

Jest to system addytywny, bez zera, oparty o wielokrotności piątki.

Cyfry posiadają następujące wartości: I = 1; V = 5; X = 10; L = 50; C = 100; D = 500; M = 1000. Liczbę tworzy ciąg cyfr zapisanych w kolejności od największej do najmniejszej. Dla uproszczenia zapisu niektórych liczb wprowadzono odstępstwo od tej zasady: cyfra mniejsza poprzedzająca większą posiada wartość ujemną. Reprezentacja większych liczb wymagała kolejnej modyfikacji: stosowano mnożenie przez 100 i 1000, oznaczając te operacje odpowiednio dwoma pionowymi kreskami po obu stronach liczby oraz kreską poziomą nad daną liczbą

Przykłady:

MCLVII = 1157

DLIX = 559

MCMIX = 1909

|MM| = 200000

## Ciekawostka:

### System Majów.

Plemię Majów opracowało system liczbowy pozycyjny na długo przed wprowadzeniem symboli arabskich w Europie. System ten opierał się na trzech symbolach: kropka, kreska i muszla:

- znak kropki oznaczał jednostkę,
- pozioma kreska oznaczała piątkę,
- muszla oznaczała zero.

Liczy zapisywano w postaci kombinacji

kropek i kresek, które odpowiednio pogrupowane stanowiły podstawowy zestaw cyfr.

0	1	2	3	4	5	6	7	8	9
	*	**	***	****	—	—*	—**	—***	—****
10	11	12	13	14	15	16	17	18	19
====	*=	**=	***=	****=	=====	====*	====**	====***	====****



Podstawowy zestaw cyfr zbudowany został w sposób addytywny, jednak system Majów był systemem pozycyjnym opierającym się na podstawie dwadzieścia. Istniał podział na jednostki wyższych rzędów:

- 1 (*kin*) – jednostka
- 20 (*unial*) - 20 x *kin*
- 360 (*tun*) - 18 x *unial*
- 7 200 (*katun*) - 20 x *tun*
- 144 000 (*baktun*) - 20 x *katun*
- 2 880 000 (*piktun*) - 20 x *baktun*

Odstępstwo od systemu dwudziestkowego przy tworzeniu wyższych rzędów (jednostka *tun* powinna wynosić 20 x *unial* = 400, a wynosiła 360) miało związek z kalendarzem Majów. Liczył on 18 miesięcy dwudziestodniowych oraz 5 dni dodatkowych co daje łącznie 365 dni.

System Majów jest bardzo dokładnym, wyrafinowanym oraz złożonym systemem liczbowym i kalendarzowym. Kropki i linie przeważnie łączone były z zestawem pięknych liczb głów w postaci glifów



*Liczbowe głowy Majów dla liczb od 0 do 19:*





## 1.1 Systemy liczbowe stosowane w informatyce - reprezentacja liczb całkowitych, znaków i napisów w komputerze.

System dziesiętny nie jest jedynym stosowanym na co dzień systemem pozycyjnym. Np. **systemu sześćdziesiątkowego** używamy do określenia czasu:

1 godzina = 60 minut;

1 minuta = 60 sekund.

**Podstawowym systemem liczbowym używanym w informatyce jest system pozycyjny o podstawie  $p=2$ .**

### System dwójkowy.

Jest to system pozycyjny, w którym  $p = 2$ , zaś do zapisu liczby wykorzystujemy dwa znaki  $\{0,1\}$ . Na końcu liczby zapisanej w tym systemie zwykle dopisuje się literę B.

Liczby w systemie dwójkowym nazywa się, wymawiając każdą cyfrę osobno. Tak więc liczbę 101B czytamy „jeden zero jeden” nie zaś „sto jeden”. Ten ostatni sposób sugerowałby, że mamy do czynienia z zapisem dziesiętnym. System dwójkowy (zwany także binarnym - z j. ang. binary) ma zastosowanie w elektronicznych układach cyfrowych z uwagi na łatwą implementację sprzętową. Dwa stany 0 i 1 o wiele łatwiej zasymulować elektrycznie, niż np. 10. Wystarczy bowiem określona wartość napięcia (czy natężenia) dla wyrażenia stanu 1 oraz jego brak dla stanu 0. Gdybyśmy chcieli zastosować system dziesiętny bezpośrednio, trzeba by aż 10 poziomów napięć!

### Wyznaczanie wartości dziesiętnej liczby zapisanej binarnie.

Aby obliczyć wartość dziesiętną liczby zapisanej binarnie, wystarczy wymnożyć cyfry liczby binarnej przez odpowiednie potęgi podstawy systemu (liczby 2), a następnie dodać wyniki.

### Przykłady:

$$111_B = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 = 1 + 2 + 4 = 7_D$$

$$10111101_B = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7$$

$$= 1 + 0 + 2 + 4 + 8 + 16 + 32 + 64 + 0 + 256 = 383_D$$

Cyfry 1 w zapisie binarnym informują, które wartości potęg liczby 2 są wykorzystywane.

W celu szybkiego przekształcania liczb binarnych na postać dziesiętną dobrze jest zapamiętać krotności poszczególnych wag systemu binarnego zamieszczone poniżej.

Zapis w systemie binarnym	Zapis w systemie dziesiętnym
1	$2^0 = 1$
10	$2^1 = 2$
100	$2^2 = 4$
1000	$2^3 = 8$
10000	$2^4 = 16$
100000	$2^5 = 32$
1000000	$2^6 = 64$
10000000	$2^7 = 128$
100000000	$2^8 = 256$
1000000000	$2^9 = 512$
10000000000	$2^{10} = 1024$

Tabela 1 Zapis binarny kolejnych potęg liczby 2.

Dzięki temu w prosty sposób możemy przekształcić liczbę binarną, sumując odpowiedniki dziesiętne wszędzie tam, gdzie w ciągu dwójkowym występują jedynki:

<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
<b>64</b>	<b>32</b>			<b>2</b>		<b>1</b>

$$64 + 32 + 2 + 1 = 99$$

W systemie dwójkowym, podobnie jak w innych systemach pozycyjnych o podstawie  $>1$ , można dopisywać 0 na początku liczby. Nie wpływa to na wartość liczby, np.:  $10_2 = 010_2 = 0010_2$



## Wyznaczanie cyfr binarnych liczby dziesiętnej.

### Sposób 1.

Aby wyznaczyć cyfry binarne liczby dziesiętnej wykorzystujemy dzielenie liczby dziesiętnej przez podstawę systemu  $p=2$ . Dzielną jest liczba dziesiętna, a dzielnikiem — podstawa systemu binarnego, czyli 2. Wynik z pierwszego dzielenia ponownie jest dzielony przez 2, i tak aż do uzyskania 0. Liczba binarna powstaje na bazie reszt zapisanych w odwrotnej kolejności:

$$\begin{array}{l}
 25 : 2 = 12 \qquad r = 1 \\
 12 : 2 = 6 \qquad r = 0 \\
 6 : 2 = 3 \qquad r = 0 \\
 3 : 2 = 1 \qquad r = 1 \\
 1 : 2 = 0 \qquad r = 1
 \end{array}$$



Tabela 2 Wyznaczanie cyfr binarnych liczby dziesiętnej.

$$25_D = 11001_2$$

W tym sposobie postępujemy wg. algorytmu:

- dziel liczbę dziesiętną przez 2 zapamiętując reszty z dzielenia.
- zapisz reszty w odwrotnej kolejności.

### Sposób 2.

Określamy, które wagi systemu (wartości potęg 2) należy wybrać, aby otrzymać liczbę dziesiętną. Np.  $81_D = 1010001_B$

	1	0	1	0	0	0	1
128	64	32	16	8	4	2	1
	64	+	16		+		1

$$64 + 16 + 1 = 81_D$$



### Ćwiczenie 1.

Podaj zapis dziesiętny liczby, która w systemie dwójkowym jest największą liczbą:

- a. ośmocyfrową,
- b. szesnastocyfrową.

### Ćwiczenie 2.

Oblicz rozwinięcie dziesiętne liczby  $10110011_2$ .

### Ćwiczenie 3.

Wyznacz cyfry binarne liczby  $212_{10}$ .



## Rozwiązania ćwiczeń.

### Ćwiczenie 1.

- a. Największą ośmiocyfrową liczbą w systemie dwójkowym jest:

$$\begin{aligned}11111111_2 &= 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255_{10}.\end{aligned}$$

- b. Największą szesnastocyfrową liczbą w systemie dwójkowym jest:

$$\begin{aligned}1111111111111111_2 &= 1 \cdot 2^{15} + 1 \cdot 2^{14} + 1 \cdot 2^{13} + 1 \cdot 2^{12} + 1 \cdot 2^{10} + 1 \cdot 2^{11} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 \\ &\quad + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 2^8 \cdot (1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) + 1 \cdot 2^7 \\ &\quad + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2^8 \cdot 255 + 255 \\ &= 256 \cdot 255 + 255 = 65535_{10}.\end{aligned}$$

### Ćwiczenie 2.

$$179_{10}$$

### Ćwiczenie 3.

$$11010100_2$$

## System szesnastkowy i ósemkowy.

Ponieważ zapis dużych liczb w systemie binarnym jest długi i mało czytelny w informatyce wykorzystuje się **system szesnastkowy** (ang. hexadecimal — **heksadecymalny**) i **system ósemkowy (oktalny)**.

Ethernetowe karty sieciowe mają 48-bitowy, unikatowy adres sprzętowy zapisany w postaci szesnastkowej, np. **00:50:56:C0:00:08**.

Oprogramowanie do wyszukiwania błędów w skompilowanych plikach binarnych przekształca pierwotny zapis danych dwójkowych na krótszy, szesnastkowy, ułatwiając w ten sposób analizę kodu. W systemie binarnym odpowiednik dziesiętnej liczby  $15_D$  ma aż cztery cyfry  $1111_B$ , natomiast w szesnastkowym — tylko jedną  $F_H$ .

Podstawę systemu heksadecymalnego stanowi 16 znaków. Liczby zapisywane są za pomocą cyfr 0, 1, ..., 9 i liter A, B, C, D, E, F.

Literom odpowiadają liczby A(10), B(11), C(12), ..., F(15).

## Konwersja liczby szesnastkowej na postać dziesiętną.

Chcąc dokonać konwersji liczby szesnastkowej na postać dziesiętną, powinniśmy użyć zapisu wielomianowego:

$$p=16, a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\},$$

$$4C5_{16} = 4 \cdot C_1 \cdot 2_0 = 4 \cdot 16^2 + C \cdot 16^1 + 2 \cdot 16^0 = 4 \cdot 256 + 12(C) \cdot 16 + 5 \cdot 1 = 1221_{10}$$

Kolejne cyfry w liczbie heksadecymalnej należy ponumerować, począwszy od pierwszej(0) z prawej strony. Następnie każdą cyfrę mnożymy przez wagę otrzymaną z podstawy (16) podniesionej do potęgi równej pozycji. Po przemnożeniu cyfr przez wagi (litery należy zamienić na odpowiedniki dziesiętne) wykonujemy sumowanie. Otrzymana liczba dziesiętna jest odpowiednikiem liczby szesnastkowej. Liczba zapisana w systemie szesnastkowym jako  $4C5_H$  odpowiada  $1221_D$  w systemie dziesiętnym.



### Konwersja liczby dziesiętnej na postać szesnastkową.

Aby dokonać zamiany liczby dziesiętnej na postać szesnastkową, należy wykonać cykliczne dzielenie z resztą. Dzielną jest liczba dziesiętna, natomiast dzielnikiem — podstawa systemu heksadecymalnego, czyli 16. Wynik uzyskany z pierwszego dzielenia ponownie jest dzielony przez 16, i tak aż do uzyskania 0. Liczba szesnastkowa powstaje na bazie reszt zapisanych w odwrotnej kolejności. Wartości powyżej 9 koduje się za pomocą odpowiednich cyfr-liter, np. A:

$$1221 : 16 = 76$$

$$r = 5$$

$$76 : 16 = 4$$

$$r = 12 \text{ (C)}$$

$$4 : 16 = 0$$

$$r = 4$$



$$1221_D = 4C5_H$$

### Konwersja liczby binarnej na postać szesnastkową i odwrotnie.

Kod heksadecymalny bardzo łatwo konwertuje się do kodu binarnego i odwrotnie.

Pomocna do tego jest poniższa tabela:

Cyfra szesnastkowa	Cyfra dwójkowa	Cyfra szesnastkowa	Cyfra dwójkowa
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111



*Konwersję liczby binarnej na postać szesnastkową* należy rozpocząć od pogrupowania ciągu po cztery cyfry. Grupowanie rozpoczynamy od prawej strony i kontynuujemy aż do uzyskania końca liczby. Jeżeli ostatnie cyfry w pogrupowanej liczbie mają mniej niż cztery znaki, należy uzupełnić puste pozycje zerami:

$$1011110101110100000101_B = 10 \mid 1111 \mid 0101 \mid 1101 \mid 0000 \mid 0101_B =$$

$$= 0010 \mid 1111 \mid 0101 \mid 1101 \mid 0000 \mid 0101_B =$$

Wszystkie pogrupowane znaki należy zamienić na odpowiadające im cyfry heksadecymalne:

0010	1111	0101	1101	0000	0101
2	F	5	D	0	5

$$1011110101110100000101_B = 2F5D05_H$$

### Konwersja z liczby szesnastkowej na binarną.

*Konwersja z liczby szesnastkowej na binarną* jest jeszcze prostsza. Wystarczy na podstawie tabeli zamienić cyfry heksadecymalne na czterocyfrowe ciągi binarne i połączyć je w jedną liczbę:

1010	0100	1011	1001	1111	0000
A	4	B	9	F	0

$$10100100101110011110000_B = A4B9F0_H$$

## System ósemkowy

System ósemkowy (ang. octal — oktalny) jest pozycyjnym systemem liczbowym, w którym podstawę stanowi osiem kolejnych cyfr arabskich: 0, 1, 2, 3, 4, 5, 6, 7. System ten jest rzadko stosowany; zastosowanie można zobaczyć w uniksowym poleceniu **chmod** (służącym do zmiany uprawnień dostępu do plików i katalogów).

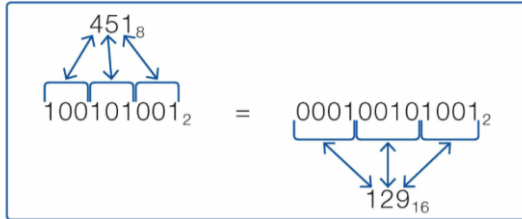




Liczba naturalna  $l_0$  w systemie ósemkowym ma postać:  $a_i \dots a_1 a_0$ , gdzie  $a_{i \dots 10}$  przyjmuje wartość 0, 1, 2, 3, 4, 5, 6, 7, np.  $212_8$ .

Konwersję liczb ósemkowych na postać dziesiętną i odwrotnie wykonuje się analogicznie jak w przykładach poświęconych systemom: binarnemu i szesnastkowemu.

### Zamiana zapisu liczby pomiędzy systemami ósemkowym i szesnastkowym.



### Ćwiczenie 1.

Zapisz liczbę:

- $10110011_2$  na liczbę w systemie szesnastkowym;
- $A9F6_{16}$  na liczbę w systemie dziesiętnym;
- $212_{10}$  na liczbę w systemie szesnastkowym;
- $548_{10}$  na liczbę w systemie ósemkowym (oktalnym);
- $12314_8$  na liczbę w systemie szesnastkowym.



## Rozwiązania ćwiczenia.

### Ćwiczenie 1.

- a) Dzielimy liczbę  $10110011_2$  na bloki czterocyfrowe (gdyż  $16 = 2^4$ ) i zamieniamy osobno na system szesnastkowy.

$$1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11_{10} = B_{16}$$

$$0011_2 = 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2 + 1 = 3_{10} = 3_{16}$$

Czyli

$$10110011_2 = B3_{16}.$$

- b)  $A9F6_{16} = 43510_{10}$ .  
c)  $212_{10} = D4_{16}$   
d)  $1044_8$ ;  
e)  $14CC_{16}$ ;

## Reprezentacja liczb całkowitych, znaków i napisów w komputerze .

### Reprezentacja liczb całkowitych w komputerze.

W reprezentacji liczby ze znakiem przeznaczamy **najbardziej znaczący bit (najstarszy bit)** na zapis znaku liczby. Jest to bit znajdujący się najbardziej z lewej strony. **Bit najmniej znaczący** to bit znajdujący się najbardziej z prawej strony

Istnieje wiele umownych sposobów zapisywania liczb ujemnych w systemie dwójkowym. Najprostszym z nich jest zapisanie bitu znaku na najbardziej znaczącej pozycji (pierwszej od lewej strony). Przyjmujemy, że liczba ujemna ma bit znaku 1, natomiast liczba dodatnia i zero ma bit znaku równy 0.

$1=0001$ ,

$-1=1001$ .

Ważnym jest, aby system zapisu liczb był ściśle określony, aby wiadomo było, iż liczby zapisujemy na przykład na czterech bitach, z których pierwszy jest bitem znaku. Nie określenie dokładnie sposoby zapisu liczb może prowadzić do błędów. Dla przykładu liczba - 1 zapisana w systemie dwójkowym na czterech bitach, gdzie pierwszy jest bitem znaku może być zinterpretowana jako liczba 9.

Taki system jednak nie jest stosowany ze względu na problemy z operacjami arytmetycznymi (np. dodanie 1 i -1 nie daje 0!)

Współcześnie przyjmuje się, że jeden bajt jest równy 8 bitom. Za pomocą bajta można przedstawić  $2^8$ , czyli 256 różnych informacji.

Ciąg n bitów pozwala zapisać  $2^n$  symboli (znaków lub liczb).

*Za pomocą 1 bajta możemy zapamiętać liczbę całkowitą bez znaku z zakresu od 0 do 255.*

Reprezentując liczby ujemne musimy 1 bit poświęcić na zapamiętanie znaku liczby.

Na 1B możemy przedstawić liczby z zakresu  **$-2^7$  do  $2^7 - 1$**  (ujemne od -128 do -1 oraz nieujemne od 0 do 127).

Jest to zbyt mały zakres więc do reprezentacji liczb całkowitych wykorzystuje się więcej niż 1 bajt.

Aby można było prawidłowo wykonywać działania na liczbach całkowitych ujemnych wprowadzono specjalne **kodowanie U2** (kod uzupełnień do 2). Zapis w U2 liczb całkowitych nieujemnych odpowiada naturalnemu zapisowi liczb binarnych.

### **Sposób na przeliczania ujemnej liczby dziesiętnej na zapis U2.**

#### **I sposób**

Liczba ujemna musi mieć ustawiony na 1 bit znaku. Zadanie sprowadza się do znalezienia wartości pozostałych bitów. Bit znaku stoi na pozycji o wadze ( $-2^{n-1}$ ),  $n$  - ilość bitów w formacie U2. Pozostałe bity zapisu liczby tworzą naturalny kod dwójkowy. Wartość tego kodu musi być taka, aby po dodaniu jej do wagi pozycji znakowej otrzymać wartość kodowanej liczby.

Zapis w formie równania:

kodowana liczba = waga bitu znakowego + wartość kodu pozostałych bitów

kodowana liczba =  $(-2^{n-1})$  + wartość kodu pozostałych bitów

stąd :

wartość kodu pozostałych bitów =  $2^{n-1}$  + kodowana liczba

Po wyznaczeniu wartości tego kodu tworzymy jego zapis w systemie dwójkowym, uzupełniamy w miarę potrzeby bitem 0 do długości formatu U2 - 1 i dodajemy bit znakowy 1.

Konwersja jest gotowa.

**Przykład:** Wyznaczyć 8-mio bitowy kod U2 dla liczby dziesiętnej  $(-45)_{10}$ .

Wyznaczamy moduł wagi pozycji znakowej.

Dla  $n = 8$   $2^{n-1} = 2^7 = 128$

wartość kodu pozostałych bitów =  $128 + (-45) = 83 = 1010011_2$

Dodajemy bit znaku równy 1 i otrzymujemy:

$(-45)_{10} = 11010011_{U2}$ .

#### **II sposób.**

Aby zapisać w kodzie U2 liczbę ujemną, należy:

najpierw dodać do niej wartość podwojonej wagi najstarszego bitu:



Jeżeli liczbę (-1) chcemy przedstawić w U2 na 8 bitach dodajemy do niej liczbę  $2^8$  (najstarszy bit ma wagę  $2^7$ , zaś podwojona wartość bitu jest równa  $2^8$  bo  $2 \cdot 2^7 = 2^8 = 256$ ):

$$-1 + 256 = 255.$$

Liczba -1 w kodzie U2 odpowiada liczbie dziesiętnej 255, czyli  $11111111_2$ .

Zatem:

$$1 = 00000001_{U2}$$

$$-1 = 11111111_{U2}$$

<b>4</b>	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	1
+	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0

Podczas dodawania pisemnego liczb 1 i -1 zapisanych w kodzie U2 **pomijamy** przeniesienie z najstarszego bitu. Otrzymujemy w ten sposób oczekiwany wynik 0.

**Liczba jest dodatnia, gdy bit znaku ma wartość 0 - suma pozostałych wag tworzy zawsze liczbę dodatnią lub zero.**

**Jeśli bit znaku przyjmie wartość 1, to liczba jest ujemna.**

W tabelce zostały przedstawione wybrane liczby całkowite ze znakiem zapisane w kodzie U2 na jednym bajcie.

Wartość dziesiętna	Zapis w kodzie U2
-128	10000000
-127	10000001
-126	10000010
...	...
-2	11111110
-1	11111111
0	00000000
1	00000001
2	00000010
...	...
125	01111101
126	01111110
127	01111111

W ogólnym przypadku, gdy liczba całkowita  $x$  zapisana jest na  $n$  bitach jej wartość dziesiętną obliczymy ze wzoru:

$$x \text{ dla } x \geq 0$$

$$X_{U2} =$$

$$2^n + x \text{ dla } x < 0$$

Obliczoną wartość dziesiętną należy zamienić na liczbę binarną.

Wartość dziesiętną liczby zapisanej na  $n$  bitach w kodzie U2 można wyznaczyć wg. schematu:

$$-b_{n-1} * 2^{n-1} + b_{n-2} * 2^{n-2} + \dots + -b_1 * 2^1 + b_0 * 2^0$$

$$10101010_2 = -1 * 2^7 + 0 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$$



## Ćwiczenia.

### Ćwiczenie 1.

Zapisz poniższe liczby dziesiętne w kodzie U2 z wykorzystaniem 1B:

- a. -100;
- b. 56;
- c. -15;

### Ćwiczenie 2.

Podaj wartość dziesiętną liczby  $11010101_{U2}$



## Rozwiązanie ćwiczenia.

### Ćwiczenie 1.

a.

Chcemy zapisać liczbę  $-100$  z użyciem jednego bajta (tj. 8 bitów) w kodzie  $U_2$ . Dodajemy zatem do niej liczbę  $2^8 = 256$ , tj.

$$-100 + 256 = 156$$

i zapisujemy w kodzie binarnym:

156 : 2	78	0
78 : 2	39	0
39 : 2	19	1
19 : 2	9	1
9 : 2	4	1
4 : 2	2	0
2 : 2	1	0
1 : 2	0	1

Otrzymujemy zatem wynik  $10011100_{U_2}$ .

b.

Liczba jest dodatnia, więc zamieniamy ją na liczbę binarną:

56 : 2	28	0
28 : 2	14	0
14 : 2	7	0
7 : 2	3	1
3 : 2	1	1
1 : 2	0	1

Aby wykorzystać pełen bajt, uzupełniamy zapis zerami do 8 cyfr i otrzymujemy

$$00111000_{U_2}.$$

c.

Chcemy zapisać liczbę  $-15$  z użyciem jednego bajta (tj. 8 bitów) w kodzie  $U_2$ . Dodajemy zatem do niej liczbę  $2^8 = 256$ , tj.

$$-15 + 256 = 241$$

i zapisujemy w kodzie binarnym:





241 : 2	120	1
120 : 2	60	0
60 : 2	30	0
30 : 2	15	0
15 : 2	7	1
7 : 2	3	1
3 : 2	1	1
1 : 2	0	1

Stąd odpowiedź  $11110001_{U_2}$ .

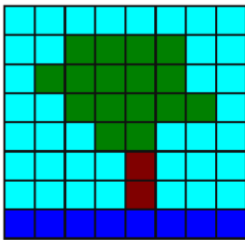
### Ćwiczenie 2.

$$11010101_{U_2} = -1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0 = -128 + 64 + 16 + 4 + 1 = -43_{10}$$



## Kodowanie grafiki.

Obrazy, dźwięki i animacje, pamiętane są w komputerze w formie cyfrowej – zer i jedynek. Załóżmy, iż chcemy zakodować binarnie obrazek pokazany poniżej: Jest on złożony z różnokolorowych punktów, które nazywamy pikselami (z języka ang. picture element – element obrazu, punkt).



Punkty są tylko w czterech kolorach. Układamy tablicę kodową kolorów, w której każdemu kolorowi punktu przyporządkujemy jeden symbol dwubitowy:

	- 00
	- 01
	- 10
	- 11

Powiązaliśmy w ten sposób informację z reprezentującymi ją symbolami. Teraz wystarczy już tylko każdy piksel zastąpić symbolem dwubitowym. W tej postaci obrazek może być przechowywany w pamięci komputera, przesyłany przez sieci teleinformatyczne oraz przetwarzany.

00	00	00	00	00	00	00	00	00	00	0000000000000000
00	11	11	11	11	00	00	00	00	00	0000111111110000
00	11	11	11	11	00	00	00	00	00	0011111111110000
00	11	11	11	11	00	00	00	00	00	0000111111111100
00	00	11	11	00	00	00	00	00	00	0000001111000000
00	00	00	10	00	00	00	00	00	00	0000000010000000
00	00	00	10	00	00	00	00	00	00	0000000010000000
01	01	01	01	01	01	01	01	01	01	0101010101010101

Zwróćmy uwagę na małą czytelność dla ludzi informacji zapisanej w systemie binarnym.



Szczególnie, jeśli wszystkie bity zapiszemy w jednym ciągu:

```
000000000000000000001111111100000011111111110000000011111111110000000011  
110000000000000000100000000101010101010101
```

System ten jest przeznaczony dla maszyn, które nie nudzą się i nie męczą.

## Reprezentacja znaków i napisów w komputerze.

Pojedyncze znaki (cyfry, litery i inne znaki) kodowane są w pamięci komputera za pomocą liczb. Kodowanie znaków to zunifikowany sposób reprezentacji znaków drukowanych i symboli sterujących w formie liczbowej, w celu uproszczenia składowania, przetwarzania i przekazywania danych w sieciach teleinformatycznych. Termin ten nie jest związany tylko z informatyką, powszechnie znany jest alfabet Morse'a wykorzystywany w telegrafii czy alfabet Braille'a umożliwiającą czytanie osobom niewidomym.

Historycznie najstarszym i jednocześnie podstawowym systemem kodowania obecnie stosowanym w komputerach jest ASCII (ang. *American Standard Code for Information Interchange*). Standard ten jest 7-bitowy co oznacza, że pozwala zakodować 128 znaków. Definiuje 33 symbole sterujące, 94 znaki drukowane i znak spacji. System ten ogranicza jednak zestaw liter do alfabetu angielskiego.

Kod dziesiętny	Znak	Kod dziesiętny	Znak	Kod dziesiętny	Znak	Kod dziesiętny	Znak
32	spacja	56	8	80	P	104	h
33	!	57	9	81	Q	105	i
34	"	58	:	82	R	106	j
35	#	59	;	83	S	107	k
36	\$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	'	63	?	87	W	111	o
40	(	64	@	88	X	112	p
41	)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	r
43	+	67	C	91	[	115	s
44	,	68	D	92	\	116	t
45	-	69	E	93	]	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95	_	119	w
48	0	72	H	96	`	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{
52	4	76	L	100	d	124	
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g	127	Del

**Tabela 3 Tablica kodów ASCII z zakresu od 32 do 127.**

Współczesne komputery używają 8-bitowych bajtów do przechowywania danych w pamięci co dało możliwość rozszerzenia zestawu ASCII o nowe elementy. Tak powstały popularne 8-bitowe rodziny kodowania ISO 8859 oraz Windows-125X. Polskie znaki diakrytyczne umieszczone zostały odpowiednio w Windows-1250 i ISO 8859-2, ostatni zestaw został objęty Polską Normą.

Jeden bajt dający zapisać 256 symboli był niewystarczający do zapisania wszystkich alfabetów używanych na świecie. Powstało więc wiele standardów, dla szeregu języków wykluczających się. Każdy z nich korzystał z tego samego zakresu liczb 0-255. Stan taki utrudniał przetwarzanie dokumentów wielojęzycznych.

W latach 90-tych zaproponowano standard Unicode będący w założeniu uniwersalnym zestawem znaków obejmującym wszystkie pisma świata. W podstawowej wersji każdy symbol zbioru reprezentowany jest jednoznacznie przez 2 bajty, co daje 65536 unikalnych znaków. Unicode nie definiuje sposobu kodowania, odpowiada tylko za "ponumerowanie" znaków. Najbardziej znanymi metodami reprezentacji są: UTF-8, UTF-16 i UTF-32.



## 1.2 Algorytm zamiany reprezentacji liczby dziesiętnej na dwójkową.

### Specyfikacja:

#### Dane:

d – liczba całkowita zapisana w postaci dziesiętnej,  
 $0 < d \leq 2^{31} - 1$

#### Wynik:

b – napis reprezentujący zapis binarny liczby d.

### Przykładowy pseudokod algorytmu:

b ← ""

#### Dopóki $d > 0$ wykonuj

jeśli  $d \bmod 2 = 0$  to b ← '0' + b

w przeciwnym przypadku b ← '1' + b

d ← d div 2

#### Wypisz b

Kolejno wyznaczone cyfry binarne łączymy w napis, aktualnie wyznaczoną cyfrę dołączamy na początek napisu złożonego z otrzymanych wcześniej cyfr. Cyfry binarne będą pamiętane jako znaki (*ujmujemy je w apostrofy*).

Wartością początkową zmiennej b jest napis pusty.



Iteracja pętli	Wartość zmiennej d	Wartość zmiennej b
–	–	""
1	26	'0'+""="0"
2	13	'1'+"0"="10"
3	6	'0'+"10"="010"
4	3	'1'+"010"="1010"
5	1	'1'+"1010"="11010"
–	0	"11010"

**Tabela 4 Działanie algorytmu dla liczby dziesiętnej 26**

## Ćwiczenie 1.

Napisz program, który po podaniu liczby całkowitej dodatniej wyświetli jej reprezentację binarną. Sprawdź działanie programu dla różnych liczb.

*Kod źródłowy programu realizującego algorytm zamiany reprezentacji liczby dziesiętnej na dwójkową.*

```
5 |
6 | #include <iostream>
7 | #include <string>
8 |
9 | using namespace std;
10 |
11 | int main()
12 | {
13 |     int d;
14 |     string b="";
15 |     cout << "Podaj liczbę dziesiętną: ";
16 |     cin >> d;
17 |     while(d > 0)
18 |     {
19 |         if(d%2 == 0) b = '0' + b;
20 |         else b = '1' + b;
21 |         d = d / 2;
22 |     }
23 |     cout << "Liczba binarna: " << b;
24 |     return 0;
25 | }
26 |
```



## 1.3 Algorytm zamiany reprezentacji liczby dwójkowej na dziesiętną.

*Specyfikacja:*

**Dane:**

b – napis reprezentujący liczbę binarną, długość napisu mniejsza od 32

**Wynik:**

d – wartość dziesiętna liczby binarnej b

potęga  $\leftarrow 1$

d  $\leftarrow 0$

dla  $i \leftarrow$  długość liczby binarnej -1, ..., 0 **wykonuj**

**jeśli** b[i]='1' to d  $\leftarrow$  d+ potęga

    potęga  $\leftarrow$  potęga \*2

**wypisz** d

Sposób I polega na sumowaniu odpowiednich potęg liczby 2. Przeglądając cyfry liczby binarnej od prawej do lewej należy wymnażać wartość zmiennej *potęga* przez 2. Jej wartość początkowa wynosi 2 ( $2^0$ ). Obliczamy w ten sposób kolejne potęgi liczby 2.

### Ćwiczenie 1.

Napisz program, który po podaniu liczby binarnej wyświetli jej reprezentację dziesiętną z wykorzystaniem dodawania odpowiednich potęg liczby 2. Sprawdź działanie programu dla różnych liczb.





*Kod źródłowy programu realizującego algorytm zamiany reprezentacji liczby binarnej na dziesiętną.*

```
4
5 #include <iostream>
6 #include <string>
7
8 using namespace std;
9
10 int main()
11 {
12     int i, d = 0, potega = 1;
13     string b;
14     cout << "Podaj liczbe binarna: ";    cin >> b;
15
16     for(i = b.size() - 1; i >= 0; i--)
17     {
18         if(b[i] == '1') d = d + potega;
19         potega = potega * 2;
20     }
21     cout << "Liczba dziesiętna: " << d;
22     return 0;
23 }
24
```

## *II sposób*

Przełamy cyfry liczby binarnej od lewej do prawej strony czyli od najbardziej znaczącej. Nie musimy w oddzielnej zmiennej wyznaczać potęg liczby 2.

## *Przykład*

$$\begin{aligned} 1101_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 2 \cdot (1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0) + 1 \\ &= 2 \cdot (2 \cdot (1 \cdot 2^1 + 1 \cdot 2^0) + 0) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot (1 \cdot 2^0) + 1) + 0) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot (1) + 1) + 0) + 1 \end{aligned}$$



*Zapis powyższego algorytmu w pseudokodzie.*

$d \leftarrow 0$

dla  $i \leftarrow 0, 1, \dots$ , długość liczby binarnej -1, ....., 0 **wykonuj**

$d \leftarrow d \cdot 2$

jeśli  $b[i] = '1'$  to  $d \leftarrow d + 1$

wypisz  $d$

Iteracja pętli	Cyfry liczby binarnej w kolejności od najbardziej znaczącej	Obliczana wartość dziesiętna
-	-	0
1	1	$0 \cdot 2 + 1 = 1$
2	1	$1 \cdot 2 + 1 = 3$
3	0	$3 \cdot 2 = 6$
4	1	$6 \cdot 2 + 1 = 13$

**Tabela 5 Wyznaczanie wartości dziesiętnej liczby  $1101_2$**

## Ćwiczenie 2

Napisz program, który po podaniu liczby binarnej wyświetli jej reprezentację dziesiętną z wykorzystaniem drugiego z opisanych algorytmów. Sprawdź działanie programu dla różnych liczb.

*Kod źródłowy programu realizującego algorytm zamiany reprezentacji liczby binarnej na dziesiętną.*

```
4
5 #include <iostream>
6 #include <string>
7
8 using namespace std;
9
10 int main()
11 {
12     int i, d = 0;
13     string b;
14     cout << "Podaj liczbę binarną: ";
15     cin >> b;
16     for(i = 0; i < b.size(); i++)
17     {
18         d = d * 2;
19         if(b[i] == '1') d++;
20     }
21     cout << "Liczba dziesiętna: " << d;
22     return 0;
23 }
24
```

## 1.4 Uogólnienie algorytmów zamiany reprezentacji liczb między systemami pozycyjnymi.

Algorytmy zamiany liczby dziesiętnej na liczbę w dowolnym systemie pozycyjnym i odwrotnie są analogiczne do omawianych powyżej. Liczbę 2 należy zastąpić podstawą systemu  $p$ . Musimy także uogólnić dołączenie aktualnie otrzymanej cyfry do wyniku pamiętając, że do zapisu liczby w danym systemie wykorzystujemy  $p$  różnych znaków.

Mając do dyspozycji 10 cyfr dziesiętnych oraz 26 wielkich liter alfabetu łacińskiego ograniczymy podstawę systemu do 36.

### *Specyfikacja:*

#### **Dane:**

$d$  – liczba całkowita zapisana w postaci dziesiętnej,  
 $0 < d \leq 2^{31} - 1$   
 $p$  – podstawa systemu  $2 \leq p \leq 36$ .

#### **Wynik:**

$s$  – napis reprezentujący zapis liczby  $d$  w systemie o podstawie  $p$ .

### *Przykładowy pseudokod algorytmu:*

$s \leftarrow ""$

#### **Dopóki $d > 0$ wykonuj**

$s \leftarrow \text{cyfra}(d \bmod p) + s$   
w przeciwnym przypadku  $b \leftarrow '1' + b$   
 $d \leftarrow d \text{ div } p$

#### **Wypisz $s$**



Jeżeli ograniczymy podstawę systemu do  $2 \leq p \leq 10$  uwzględnimy kody ASCII cyfr od 0 do 9:

kod znaku **0** – 48;

kod znaku **1** – 49 itd.

Do wartości wyrażenie  $d \bmod p$  wystarczy dodać 48, aby otrzymać kod ASCII odpowiedniej cyfry.

## Ćwiczenie 1

Napisz program, który po podaniu liczby dziesiętnej wyświetli jej reprezentację w systemie pozycyjnym o podstawie z zakresu od 2 do 36. Sprawdź działanie programu dla różnych liczb.

*Kod źródłowy programu realizującego algorytm zamiany reprezentacji liczby binarnej na dziesiętną.*

```
9
10 int main()
11 {
12     int d, p, cyfra;
13     string s = "";
14     cout << "Podaj podstawę systemu: ";
15     cin >> p;
16
17     cout << "Podaj liczbę dziesiętną: ";
18     cin >> d;
19
20     while(d > 0)
21     {
22         cyfra = d % p;
23         if(cyfra < 10){
24             s = char('0' + cyfra) + s;
25         }
26         else{
27             s = char('A' - 10 + cyfra) + s;
28         }
29         d = d / p;
30     }
31
32     cout << "Liczba w systemie o podstawie " << p << ": " << s;
33
34     return 0;
35 }
36
```

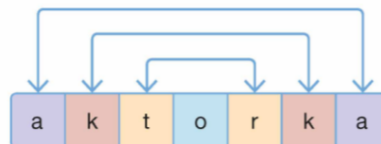
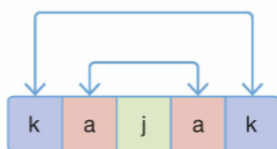
W linii 27 od kodu ASCII litery „A” odejmujemy 10, ponieważ w systemach pozycyjnych cyfry oznaczone literami zaczynają się od 10.



## 1.5 Sprawdzamy czy wyraz jest palindromem.

Palindrom (gr. palindromos – biegnący na powrót, palindromeo – biec z powrotem) – wyrażenie brzmiące tak samo czytane od lewej do prawej i od prawej do lewej. Przykładem palindromu jest: Kobyła ma mały bok. Współcześnie palindromy pełnią funkcję gry słownej. Prawdopodobnie tak było również i w przeszłości, choć pewne znaleziska sugerują, że palindromy mogły też mieć znaczenie magiczne. Przykładowe wyrazy będące palindromami: Abba, Ada, Aga, Ala, anilina, Anna, Ała, apokopa, Ara, Aviva, Axa, Bob, bób, Elle, gag, inni, łajał, łapał, łamał, łatał, kajak, kok, Malajalam, macam,

Podczas sprawdzania porównujemy kolejne pary liter: pierwszą z ostatnią, drugą z przedostatnią, itd. Znalezienie pierwszej pary różnych liter oznacza, że wyraz nie jest palindromem i możemy przerwać sprawdzanie. Jeżeli nie znajdziemy pary różnych liter to oznacza, że wyraz jest palindromem.



Nie musimy porównywać środkowej litery samej ze sobą, gdy wyraz składa się z nieparzystej liczby liter.

Wykonujemy  $n \div 2$  porównań par liter jeśli wyraz jest palindromem.

### Specyfikacja:

#### Dane:

Wyraz – słowo złożone z małych liter alfabetu łacińskiego.

#### Wynik:

Słowo TAK, gdy wyraz jest palindromem, słowo NIE w przeciwnym przypadku.

### Przykładowy pseudokod algorytmu:

Palindrom  $\leftarrow$  **prawda**

$i \leftarrow 0$

$j \leftarrow$  długość wyrazu  $- 1$

**dopóki** palindrom **oraz** ( $i < j$ ) **wykonuj**

**jeśli** wyraz [ $i$ ] = wyraz [ $j$ ] **to**

$i \leftarrow i + 1$

$j \leftarrow j - 1$

**w przeciwnym przypadku** palindrom  $\leftarrow$  **falsz**

**jeśli** palindrom **to wypisz** „TAK”

**w przeciwnym przypadku wypisz** „NIE”

Zakładamy, że badany wyraz jest palindromem więc zmiennej typu logicznego *palindrom* przypisujemy wartość **prawda**. Zmienna  $i$  wskazuje na kolejne litery od początku wyrazu (wartość początkowa jest równa 0), zaś zmienna  $j$  kolejne znaki od końca wyrazu.

Jeśli w aktualnie porównywanej parze są takie same znaki to wartość zmiennej  $i$  jest powiększana o 1, a zmiennej  $j$  pomniejszana o 1.

W przypadku gdy znaki badanej pary są różne, zmienna *palindrom* przyjmuje wartość **falsz**, co kończy wykonywanie pętli.

Jeśli wyraz jest palindromem to pętla jest wykonywana dopóki  $i < j$ .

### Ćwiczenie 1

Napisz program sprawdzający:

- czy podany wyraz złożony z małych liter alfabetu łacińskiego jest palindromem;
- zmodyfikuj program tak, aby wypisywał komunikat „TAK” bez względu na to czy do zapisu użyto dużych czy małych liter.

Fragment kodu źródłowego programu sprawdzającego czy wyraz jest palindromem.

a)

```
9
10 int main()
11 {
12     string wyraz;
13     int i = 0, j;
14     bool palindrom = true;
15     cout << "Wpisz wyraz: ";
16     cin >> wyraz;
17
18     j = wyraz.size()-1;
19
20     while(palindrom && i<j)
21     {
22         if(wyraz[i] == wyraz[j])
23         {
24             i++;
25             j--;
26         }
27         else
28             palindrom = false;
29
30     }
31
32     if(palindrom)
33         cout << "TAK";
34     else
35         cout << "NIE";
36
37     return 0;
38 }
39
40
41
42
43
44
```

b)

Modyfikujemy kod w 22 linii. Zastosujemy funkcję **toupper**. Jej wartością jest kod ASCII wielkiej litery, jeżeli parametr był małą literą. W przeciwnym przypadku parametr pozostaje bez zmian.

`toupper(wyraz[i]) == toupper(wyraz[j])`

Możemy skorzystać, także z funkcji **tolower**. Jej wartością jest kod ASCII małej litery, gdy parametr był wielką literą.

Funkcje **toupper** i **tolower** są zdefiniowane w bibliotece `cctype`.



## 1.6 Sprawdzamy czy dwa wyrazy są anagramami.

**Anagram** to wyraz lub zdanie powstałe po przestawieniu liter innego wyrazu (zdania). Oba wyrazy składają się z tej samej liczby znaków, tylko w innym porządku.

Kilka przykładów:

adam i dama

fraktal i kartafi

matura i trauma

arbuz i burza

10112 i 21011

Chcąc sprawdzić, czy dwa wyrazy to anagramy, już na samym początku należy porównać ich długość. Jeśli długości są różne to wyrazy nie są swoimi anagramami.

*Specyfikacja:*

**Dane:**  $s_1, s_2$  – napisy złożone z wielkich liter alfabetu łacińskiego,

**Wynik:** wartość logiczna **prawda**, jeśli  $s_1$  i  $s_2$  są anagramami lub **fałsz** w przeciwnym przypadku.

*Algorytm wykorzystujący sortowanie.*

Aby stwierdzić czy dwa słowa są anagramami, można posortować alfabetycznie litery w obydwu napisach, a następnie porównać znaki na tych samych pozycjach w obu wyrazach.

*Przykładowy pseudokod algorytmu:*

```
funkcja Anagramy(s1,s2)
    jeśli długość s1!= s2 to
        Zwróć fałsz i zakończ
Sortuj(s1)
Sortuj(s2)
zwróć s1 = s2 i zakończ
```

Wynikiem funkcji jest rezultat porównania napisów – wartość logiczna.

## Ćwiczenie 1

Napisz program wczytujący dwa słowa i wypisujący "TAK", gdy podane słowa są anagramami, "NIE" - w przeciwnym wypadku. Zastosuj algorytm wykorzystujący sortowanie.

*Fragment kodu źródłowego programu sprawdzającego czy dwa wyrazy są anagramami – funkcja sortująca z wykorzystaniem algorytmu sortowania przez wybieranie.*

```
5
6 #include <iostream>
7 #include <string>
8
9 using namespace std;
10
11 void Sortuj(string &s)
12 {
13     int i, j, m;
14     char ch;
15
16     for(i = 0; i<s.size()-1; i++)
17     {
18         m = i;
19         for(j = i+1; j < s.size(); j++)
20             if(s[j] < s[m])
21                 m = j;
22
23         ch = s[i];
24         s[i] = s[m];
25         s[m] = ch;
26     }
27 }
28
```

*Fragment kodu źródłowego programu sprawdzającego czy dwa wyrazy są anagramami funkcja porównująca znaki napisów.*

```
28
29 bool Anagramy(string s1, string s2)
30 {
31     if(s1.size() != s2.size())
32         return false;
33
34     Sortuj(s1);
35     Sortuj(s2);
36
37     return (s1==s2);
38 }
39
```

### *Uwaga do funkcji Sortuj:*

Domyślnie parametry typu string przekazywane są przez wartość. Operacje wewnątrz funkcji wykonywane są na kopii parametru. Zmiany nie są zachowywane po zakończeniu działania funkcji. Aby zachować zmiany, należy przekazać parametr przez referencję:

Void Sortuj(string &s)

### *Algorytm wykorzystujący zliczanie.*

Drugim sposobem sprawdzenia czy dwa słowa są anagramami, jest zliczenie występowania poszczególnych litery w obu napisach, a następnie porównanie liczby tych wystąpień.

### *Przykładowy pseudokod algorytmu:*

**funkcja** Anagramy(s1,s2)  
    **jeśli** długość s1!= s2 **to**  
        **Zwróć fałsz i zakończ**  
Zlicz(s1,lle1)  
Zlicz(s2,lle2)  
**dla** i ← 0, 1, ..., długość alfabetu -1 **wykonuj**  
    **jeśli** lle1[i] != lle2[i] **to**  
        **Zwróć fałsz i zakończ**  
**zwróć prawda i zakończ**

Zadaniem funkcji **Zlicz** jest wyznaczenie liczby wystąpień poszczególnych znaków w danym napisie (parametr pierwszy). Wynikiem działania funkcji jest tablica (parametr drugi), w której na kolejnych pozycjach przechowywane są liczby wystąpień kolejnych liter alfabetu. Tablica składa się z 26 elementów zgodnie ze specyfikacją w wyrazach mogą wystąpić tylko wielkie litery alfabetu łacińskiego. Na początku należy wyzerować wartości elementów tablicy.

Do liter słowa odwołujemy się za pośrednictwem kodów ASCII tablice zaś są indeksowane od zera. Musimy więc przesunąć wartość indeksu tak, aby element zerowy tablicy odpowiadał literze A.

*Przykładowy pseudokod algorytmu:*

**funkcja** Zlicz(s,Ile)

**dla**  $i \leftarrow 0, 1, \dots$ , długość alfabetu -1 **wykonuj**  
    Ile[i]  $\leftarrow 0$

**dla**  $i \leftarrow 0, 1, \dots$ , długość alfabetu -1 **wykonuj**  
    powiększ ile[kodAscii(s[i])- kod ASCII('A')]

## Ćwiczenie 2

Napisz program wczytujący dwa słowa i wypisujący "TAK", gdy podane słowa są anagramami, "NIE" - w przeciwnym wypadku. Zastosuj algorytm wykorzystujący zliczanie.

*Fragment kodu źródłowego programu sprawdzającego czy dwa wyrazy są anagramami funkcja zliczająca znaki w napisie oraz porównująca otrzymane wyniki.*

```
10
11 void Zlicz(string s, int Ile[])
12 {
13     for(int i = 0; i < 26; i++)
14         Ile[i] = 0;
15
16     for(int i = 0; i < s.size(); i++)
17         Ile[s[i]-'A']++;
18 }
```



```
19
20 bool Anagramy(string s1, string s2)
21 {
22     int Ile1[26], Ile2[26];
23
24     if(s1.size() != s2.size())
25         return false;
26
27     Zlicz(s1, Ile1);
28     Zlicz(s2, Ile2);
29
30     for(int i = 0; i < 26; i++)
31         if(Ile1[i] != Ile2[i])
32             return false;
33
34     return true;
35 }
36
```

### *Złożoność czasowa omówionych algorytmów.*

Złożoność pierwszego algorytmu zależy od użytej metody sortowania. Zastosowano sortowanie przez wybieranie, więc złożoność tego algorytmu jest **kwadratowa** względem długości słowa.

Drugi algorytm trzykrotnie przegląda tablicę z licznikami i dwukrotnie słowo. Jego złożoność jest **liniowa** względem większej z wartości: długość słowa, liczba liter alfabetu.

Dla krótszych słów algorytm pierwszy wykona mniej operacji.



## 2 Metody sortowania prostego.

**Sortowanie** (ang. sorting), porządkowanie - polega na ustawieniu danych lub informacji według określonego kryterium, zwykle w kolejności rosnącej (tj. od najmniejszego elementu do największego) lub malejącej (czyli odwrotnie).

Uporządkowanie zbioru danych (np. bazy danych) lub informacji znacznie upraszcza i przyspiesza przeszukiwanie takich zbiorów w poszukiwaniu określonych elementów.

Problem sortowania można zdefiniować następująco:

Danymi wejściowymi jest ciąg  $n$  liczb.

Wynikiem jest taka ich permutacja (czyli zmiana kolejności), że tworzą one ciąg rosnący (niemalejący) lub malejący (nierosnący).

Zadaniem algorytmu sortowania jest takie przestawienie elementów danego ciągu, aby były one uporządkowane rosnąco (niemalejąco).

Sortowanie jest jednym z najczęściej rozwiązywanych problemów informatycznych. Według różnych autorów, komputery spędzają od 25 do 80 procent czasu na porządkowaniu informacji. Porządek wśród elementów ułatwia i przyspiesza wykonywanie innych operacji (np. przeszukiwania).

Sortowanie jest przykładem problemu, który może być rozwiązany na wiele sposobów, a ich efektywność jest istotnie różna. Za efektywność algorytmów sortujących przyjmuje się liczbę porównań wykonywanych między elementami danych. Zwykle jest ona podawana jako zależność od liczby elementów do uporządkowania.

W tej części skryptu są opisane wybrane algorytmy sortowania.

## 2.1 Sortowanie bąbelkowe i przez wybieranie.

### Sortowanie bąbelkowe

Sortowanie bąbelkowe ciągu  $n$  liczb polega na porównywaniu parami kolejnych liczb i przestawianiu ich jeżeli występują w niewłaściwej kolejności. Nazywane jest **sortowaniem przez prostą zamianę**.

Metoda ta polega na porównywaniu ze sobą sąsiadujących elementów. Jeżeli są ustawione w niewłaściwej kolejności, to zamieniamy je miejscami.

Jeżeli chcemy posortować zbiór niemalejąco, a pierwszy element jest większy od drugiego to zamieniamy ich pozycje.

Nazwa **sortowanie bąbelkowe** (*ang. Bubble sort*) powstała przez analogie do zjawiska wypływania na powierzchnię wody bańki powietrza. Bańka o największej objętości wypływa pierwsza. Podczas sortowania największy element zbioru również wypływa, czyli jest ustawiany na właściwym miejscu.

Elementy tablicy 5, 3, 8, 1, 4 posortujemy niemalejąco algorytmem z zastosowaniem sortowania bąbelkowego:

### Krok 1

5	3	8	1	4
---	---	---	---	---

Porównujemy pierwszy element z drugim:  $5 > 3$  - zamieniamy liczby miejscami.

3	5	8	1	4
---	---	---	---	---

Porównujemy drugi element z trzecim  $5 < 8$ , - nie zamieniamy liczb miejscami.

3	5	8	1	4
---	---	---	---	---

Porównujemy trzeci element z czwartym  $8 > 1$ , - zamieniamy liczby miejscami.

3	5	1	8	4
---	---	---	---	---

Porównujemy czwarty element z piątym  $8 > 4$ , - zamieniamy liczby miejscami.

3	5	1	4	8
---	---	---	---	---

Po czterech porównaniach największy element; tablicy (liczba 8) znalazł się na ostatniej pozycji.

W **kroku 2** umieścimy właściwy element na przedostatniej pozycji.

Przy porównaniach pomijamy ostatni element tablicy, ponieważ jest już na właściwym miejscu.

3	1	4	5	8
---	---	---	---	---

W **kroku 3** przy porównaniach pomijamy dwa ostatnie elementy tablicy,

1	3	4	5	8
---	---	---	---	---

W **kroku 4** —trzy ostatnie itd.

1	3	4	5	8
---	---	---	---	---

W wyniku działania algorytmu otrzymujemy przesortowaną niemalejąco tablicę:

1	3	4	5	8
---	---	---	---	---

*Pseudokod algorytmu porządkującego elementy tablicy niemalejąco metodą sortowania bąbelkowego:*

```
dla  $i \leftarrow 1, 2, \dots, n - 1$  wykonuj
  dla  $j \leftarrow 0, 1, \dots, n - i - 1$  wykonuj
    jeśli  $A[j] > A[j+1]$  to
      pom  $\leftarrow A[j]$ 
       $A[j] \leftarrow A[j+1]$ 
       $A[j+1] \leftarrow pom$ 
```

Jedno wykonanie zewnętrznej pętli wyznacza największy element badanego ciągu - wypłyniecie jednego „bąbelka”. Ciągu jednoelementowego nie musimy sortować, dlatego liczba powtórzeń zewnętrznej pętli jest o jeden mniejsza od liczby elementów tablicy.



Liczba powtórzeń wewnętrznej pętli zależy od wartości zmiennej sterującej zewnętrznej pętli. W kolejnych iteracjach zewnętrznej pętli, liczba powtórzeń wewnętrznej pętli zmniejsza się o jeden. Do zamiany miejscami dwóch liczb wykorzystujemy zmienną pomocniczą **pom**.

## Ćwiczenie 1

Napisz i przetestuj program sortujący niemalejąco tablice liczb całkowitych algorytmem sortowania bąbelkowego.

*Fragment kodu źródłowego programu sortującego niemalejąco tablice liczb - funkcja realizująca algorytm sortowania bąbelkowego.*

```
27
28 void Sortuj(int A[])
29 {
30     int i, j, pom;
31     for(i = 1; i < N; i++)
32     {
33         for(j = 0; j < N-i; j++)
34         {
35             if(A[j] > A[j+1])
36             {
37                 pom = A[j];
38                 A[j] = A[j+1];
39                 A[j+1] = pom;
40             }
41         }
42     }
43 }
44
```

## Sortowanie przez wybieranie.

Sortowanie przez przez wybieranie polega na wyszukaniu elementu, który powinien się znaleźć na danej pozycji, i zamianie miejscami tego elementu z elementem znajdującym się dotychczas na tej pozycji.

Jeśli chcemy posortować tablicę niemalejąco, w pierwszej iteracji poszukujemy najmniejszej liczby w całej tablicy i zamieniamy ją miejscami z pierwszym elementem tablicy. W drugiej iteracji poszukujemy liczby, która powinna się znaleźć w tablicy jako druga. Pomijamy pierwszy element tablicy, ponieważ jest już na właściwym miejscu. Po znalezieniu szukanej liczby zamieniamy ją miejscami z drugim elementem tablicy. W kolejnych kro-

kach postępujemy podobnie. Może się zdarzyć, że kolejny poszukiwany element jest już na właściwej pozycji.

Elementy tablicy 5, 3, 8, 1, 4 posortujemy niemalejąco algorytmem sortowania przez wybieranie:

### Krok 1

5	3	8	1	4
---	---	---	---	---

Szukamy najmniejszego elementu w całej tablicy. Jest nim liczba 1. Zamieniamy ją miejscami z pierwszą liczbą w tablicy - liczbą 5.

### Krok 2

1	3	8	5	4
---	---	---	---	---

Szukamy najmniejszego elementu tablicy, pomijając element pierwszy. Szukaną liczbą jest 3 - znajduje się ona na właściwej pozycji.

### Krok 3

1	3	8	5	4
---	---	---	---	---

Pomijamy pierwszy i drugi element tablicy i szukamy najmniejszej wśród pozostałych liczb - jest to liczba 4. Zamieniamy ją miejscami z trzecim elementem tablicy — liczbą 8.

### Krok 4

1	3	4	5	8
---	---	---	---	---

Pomijamy trzy pierwsze elementy tablicy i szukamy liczby na przedostatnia pozycji. Jest to liczba 5, która znajduje się już na właściwym miejscu.

Wyznaczenie przedostatniego elementu wskazuje jednocześnie element na ostatnia pozycje.

Wynikiem działania algorytmu jest tablica:

1	3	4	5	8
---	---	---	---	---

*Pseudokod algorytmu porządkującego elementy tablicy niemalejąco metodą sortowania przez wybieranie:*

```
dla i ← 1, 2, ..., n - 2 wykonuj  
  m ← i  
  dla j ← i + 1, i + 2, ..., n - 1 wykonuj  
    jeśli A [j] < A [m] > to m ← j  
  pom ← A [i]  
  A [i] ← A [m]  
  A [m] ← pom
```

W każdej iteracji zewnętrznej pętli jest znajdowany najmniejszy element badanego ciągu. Liczba powtórzeń wewnętrznej pętli zależy od wartości zmiennej sterującej zewnętrzną pętlą. Rozpoczynamy od porównania elementu znajdującego się na pozycji, na którą wybierana jest liczba, z następnym elementem. Czasami liczba jest zamieniana sama ze sobą.

## Ćwiczenie 2

Napisz i przetestuj program sortujący niemalejąco tablice liczb całkowitych algorytmem sortowania przez wybieranie.

*Fragment kodu źródłowego programu sortującego niemalejąco tablice liczb - funkcja realizująca algorytm sortowania przez wybieranie.*

```
27
28 void Sort_wybor(int A[])
29 {
30     int i, j, m, pom;
31     for(i = 0; i < N-1; i++)
32     {
33         m = i;
34         for(j = i+1; j < N; j++)
35         {
36             if(A[j] < A[m])
37                 m = j;
38         }
39         pom = A[i];
40         A[i] = A[m];
41         A[m] = pom;
42     }
43 }
44
45
```

## 2.2 Sortowanie szybkie.

Jednym z podstawowych sposobów rozwiązywania problemów algorytmicznych jest metoda „dziel i zwyciężaj” (ang. divide and conquer).

Polega ona na tym, że problem dzieli się na podproblemy tego samego typu dla mniejszej liczby danych.

Sortowanie szybkie (ang. quicksort) polega na podzieleniu danych na dwa takie fragmenty, aby każdy element pierwszego fragmentu był mniejszy lub równy każdemu elementowi drugiego fragmentu. Podział ten trzeba zaplanować tak, żeby do jego przeprowadzenia wystarczyło jednokrotne przejście danych.

Dane dzielimy na fragmenty względem jednego wyróżnionego elementu, który nazywamy elementem dzielącym lub pivotem.

61	43	56	10	77	51	64	78	55	36	15
----	----	----	----	----	----	----	----	----	----	----

*Dane wraz z wyróżnionym elementem dzielącym pivotem - jest nim element środkowy.*

Liczby ze zbioru na mniejsze od 51 i większe od 51. Pivot może trafić do dowolnego z tych fragmentów.

Kroki prowadzące do wyznaczenia dwóch fragmentów:

- w pierwszym znajdują się liczby mniejsze lub równe elementowi dzielącemu,
- w drugim liczby od niego większe.

15	43	56	10	77	51	64	78	55	36	61
----	----	----	----	----	----	----	----	----	----	----

15	43	56	10	77	51	64	78	55	36	61
----	----	----	----	----	----	----	----	----	----	----

61	43	36	10	77	51	64	78	55	56	61
----	----	----	----	----	----	----	----	----	----	----



Przeglądamy elementy od lewej strony i poszukujemy elementu większego lub równego elementowi dzielącemu, a pomijamy (pozostawiamy na miejscu) elementy od niego mniejsze. Znalezione elementy przenosimy na prawą stronę. Następnie z prawej strony poszukujemy elementu mniejszego lub równego elementowi dzielącemu. Jeśli go znajdziemy, zamieniamy go miejscami z elementem znalezionym po lewej stronie. Takich par do zamiany poszukujemy tak długo, aż indeksy wskazujące elementy od lewej i prawej strony się miną.

Element dzielący jest jednocześnie wartownikiem. Zabezpiecza przed przekroczeniem zakresu tablicy, jeśli nie ma już elementu większego od wartownika przy przeglądaniu z lewej strony lub mniejszego od wartownika przy przeglądaniu z prawej strony.

61	43	36	10	51	77	64	78	55	56	61
----	----	----	----	----	----	----	----	----	----	----

Element dzielący może trafić do jednego z fragmentów (w przykładzie trafia do lewego) lub nie trafić do żadnego — wówczas jest już na właściwym miejscu.

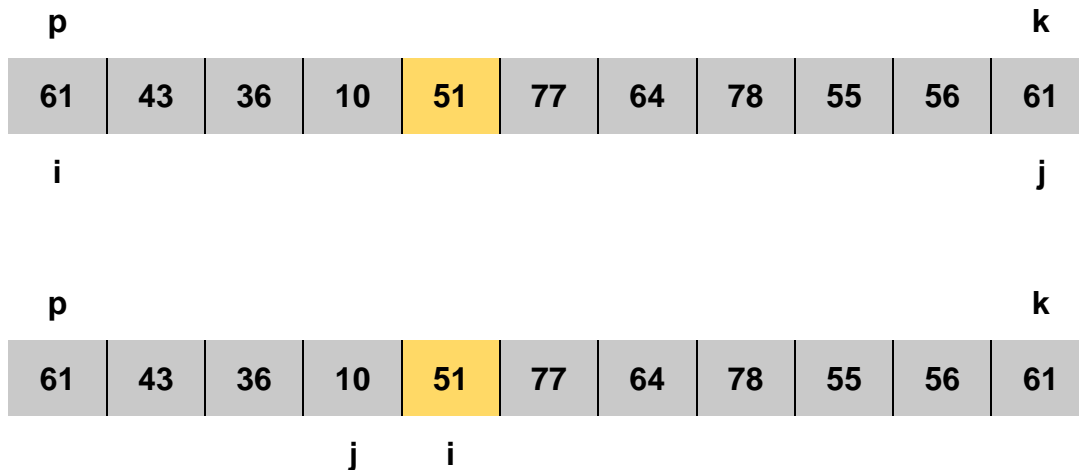
W algorytmie sortowania szybkiego będziemy wielokrotnie dzielić coraz mniejsze fragmenty tablicy.

*Zapis w pseudokodzie algorytmu podziału fragmentu tablicy:*

```
i ← p
j ← k
m ← A [(p+k) div 2] //element dzielący
dopóki i <= j wykonuj
    dopóki A [i] < m wykonuj i ← i + 1
    dopóki A [j] > m wykonuj j ← j - 1
    jeśli i <= j to
        Zamień (A[i],A[j])
        i ← i + 1
        j ← j - 1
```



*Schemat podziału fragmentu tablicy na dwa mniejsze fragmenty. Zwróć uwagę na położenie indeksów i oraz j.*



Dalsze postępowanie po podziale jest już proste: otrzymane fragmenty dzielimy tak długo, aż otrzymamy fragmenty jednoelementowe. Ponieważ każdy element z lewego fragmentu jest mniejszy lub równy każdemu elementowi z prawego fragmentu, całość jest już uporządkowana od najmniejszego elementu do największego.

Kolejne fragmenty możemy sortować (dzielić) rekurencyjnie, pod warunkiem, że  $p < j$  dla lewego fragmentu (składa się on z więcej niż jednego elementu) oraz  $i < k$  dla prawego fragmentu.

*Specyfikacja problemu sortowania szybkiego:*

**Dane:**  $A[0..n-1]$  — tablica n liczb całkowitych.

**Wynik:** liczby w tablicy A spełniające warunek  $A[0] \leq A[1] \leq \dots \leq A[n-1]$ .

### *Pseudokod algorytmu sortowania szybkiego:*

funkcja QuickSort(A[],p,k)

```
i ← p
j ← k
m ← A [(p+k) div 2]
dopóki i <= j wykonuj
  dopóki A[i] < m wykonuj i ← i + 1
  dopóki A[j] > m wykonuj j ← j - 1
  jeśli i <= j to
    Zamień(A[i],A[ j])

    I ← i + 1

    J ← j - 1
```

**jeśli** p < j **to** QuickSort(A,p,j)

**jeśli** i < k **to** QuickSort(A,i,k)

### **Ćwiczenie 1**

Napisz i przetestuj program sortujący niemalejąco tablice liczb całkowitych algorytmem QuickSort.

*Fragment kodu źródłowego programu sortującego niemalejąco tablice liczb - funkcja realizująca algorytm sortowania przez wybieranie.*

```
25
26 void QuickSort(int A[], int p, int k
27 {
28     int i = p, j = k, m = A[(p + k) /
29
30     while(i <= j)
31     {
32         while(A[i] < m) i++;
33         while(A[j] > m) j--;
34         if(i <= j)
35         {
36             swap(A[i], A[j]);
37             i++;
38             j--;
39         }
40     }
41
42     if(p < j)
43         QuickSort(A,p,j);
44
45     if(i < k)
46         QuickSort(A,i,k);
47 }
48
```



## Złożoność obliczeniowa algorytmu sortowania szybkiego

Złożoność pesymistyczna algorytmu - najmniej korzystny byłby podział na jeden element i pozostałe. Jeśli występowałyby za każdym razem (np. gdyby wartość elementu dzielącego była równa wartości minimum lub maksimum w każdym fragmencie), to jednokrotne przejście danych w tablicy powodowałoby ustawienie na właściwej pozycji tylko jednego elementu. Wówczas działanie algorytmu nie różniłoby się od metod sortowania prostego.

Pesymistyczna złożoność czasowa algorytmu sortowania szybkiego wynosi więc  $O(n^2)$ .

Prawdopodobieństwo, że tak się będzie działo dla losowych danych, szczególnie składających się z większej liczby elementów, jest bliskie zeru. **Średnia złożoność** czasowa algorytmu sortowania szybkiego wynosi  $O(n * \log n)$  i jest to jeden z najszybszych algorytmów sortujących.

Algorytmy sortowania prostego odmówione w temacie 2.1 nie wymagały użycia dodatkowej pamięci zależnej od rozmiaru danych (np. wykorzystania tablicy pomocniczej), aby te dane uporządkować. Mówimy, że takie algorytmy sortują w miejscu (in situ).

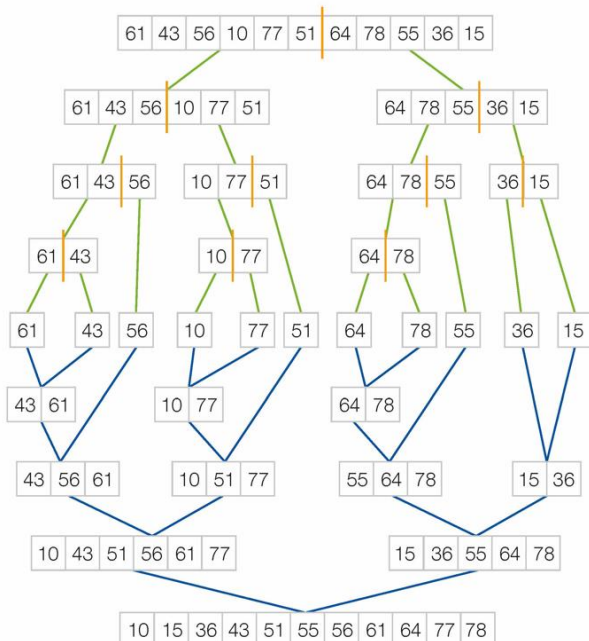
Czy algorytm sortowania szybkiego także sortuje w miejscu, a więc potrzebuje dodatkowej pamięci rzędu  $O(1)$ ? Nie korzysta on z tablicy pomocniczej, ale wywołania rekurencyjne wymagają dodatkowej pamięci w zależności od rozmiaru sortowanych danych. Średnio wywołania rekurencyjne potrzebują  $O(\log n)$  dodatkowej pamięci. Nie jest to złożoność stała  $O(1)$ , więc **nie jest to algorytm sortujący w miejscu**.



## 2.3 Sortowanie przez scalanie.

Sortowanie przez scalanie (ang. merge sort) działa na podobnej zasadzie jak algorytm sortowania szybkiego. Dane dzielimy na dwa fragmenty, ale o takiej samej liczbie elementów (z dokładnością do 1). Nie nakładamy przy tym żadnych warunków na dane podczas podziału. Po posortowaniu obydwu fragmentów danych scalamy je, czyli łączymy w całość, która będzie uporządkowana. Odpowiednie scalanie fragmentów jest kluczowym etapem algorytmu.

### *Działanie algorytmu sortowania przez scalanie.*



Pomarańczowe linie wyznaczają miejsca podziału. Podziały przeprowadzamy tak długo, aż otrzymamy fragmenty jednoelementowe (fragment złożony z jednego elementu jest uporządkowany). Zielone linie wskazują etap podziałów, a granatowe — etap scalania krótszych fragmentów w dłuższe, już uporządkowane.



*Pseudokod algorytmu sortowania przez scalanie – funkcja rekurencyjna:*

funkcja MergeSort(A[],p,k)

**jeśli**  $p < k$  **to**

$s \leftarrow (p+k) \text{ div } 2$  // indeks środka fragmentu

MergeSort(A,p,s) // sortuj lewy fragment

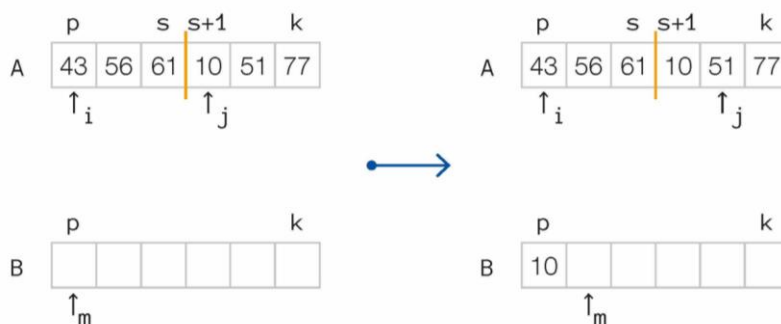
MergeSort(A,s+1,k) // sortuj prawy fragment

Merge(A,p,s,k) //scalaj posortowane fragmenty

**Scalanie posortowanych fragmentów**

Głównym zadaniem tego algorytmu jest odpowiednie scalenie dwóch fragmentów danych. Jedną z możliwości jest scalenie dwóch fragmentów tablicy (nazwijmy ją A) w tablicy pomocniczej (nazwijmy ją B), a następnie przekopiowanie scalonego fragmentu z powrotem do sortowanej tablicy. Porównywane są kolejne elementy fragmentów i mniejszy z nich jest kopiowany do tablicy pomocniczej. Indeks skopiowanego elementu przesuwany jest do następnego elementu.

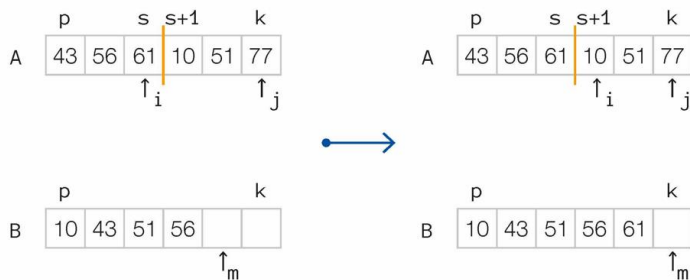
*Pierwszy krok scalania dwóch fragmentów z wykorzystaniem tablicy pomocniczej B - porównanie pierwszych elementów fragmentów i skopiowanie mniejszego.*



W ten sposób można postępować, dopóki w obydwu fragmentach są jeszcze elementy do porównywania. W pewnym momencie jeden z fragmentów zostanie w całości skopiowany do tablicy B. Wówczas do tablicy B należy skopiować brakującą część drugiego fragmentu.



Porównanie ostatnich elementów pierwszego i drugiego fragmentu



Pseudokod algorytmu scalania.

$i \leftarrow p$

$j \leftarrow s + 1$

$m \leftarrow p$

**dopóki**  $i \leq s$  oraz  $j \leq k$  **wykonuj**

**jeśli**  $A[i] < A[j]$  **to**

$B[m] \leftarrow A[i]$

$i \leftarrow i + 1$

**w przeciwnym przypadku**

$B[m] \leftarrow A[j]$

$j \leftarrow j + 1$

$m \leftarrow m + 1$

**dopóki**  $i \leq s$  **wykonuj**

$B[m] \leftarrow A[i]$

$i \leftarrow i + 1$

$m \leftarrow m + 1$

**dopóki**  $j \leq k$  **wykonuj**

$B[m] \leftarrow A[j]$

$j \leftarrow j + 1$

$m \leftarrow m + 1$

**dla**  $i \leftarrow p, \dots, k$  **wykonuj**  $A[i] \leftarrow B[i]$

Druga i trzecia pętla **dopóki** w powyższym algorytmie wykona się tylko jedna z nich, ponieważ spełniony będzie tylko jeden z warunków:  $i \leq s$  albo  $j \leq k$ . Wówczas zostanie przekopiowana brakująca końcówka jednego z fragmentów.

*Funkcja Merge, realizująca powyższy algorytm zapisany w pseudokodzie, może wyglądać następująco:*

```
25
26 void Merge(int A[], int p, int s, int k)
27 {
28     int i = p, j = s+1, m = p;
29     int B[N];
30
31     while(i <= s && j <= k)
32     {
33         if(A[i] < A[j])
34         {
35             B[m] = A[i];
36             i++;
37         }
38         else
39         {
40             B[m] = A[j];
41             j++;
42         }
43         m++;
44     }
45
46     while(i <= s)
47     {
48         B[m] = A[i];
49         i++;
50         m++;
51     }
52
53     while(j <= k)
54     {
55         B[m] = A[j];
56         j++;
57         m++;
58     }
59
60     for(i = p; i <= k; i++)
61         A[i] = B[i];
62 }
```

Zwróć uwagę na deklarację tablicy pomocniczej B (linia 29). Przechowuje ona elementy z tablicy A od indeksu p do k, czyli  $k - p + 1$  elementów. N jest stałą zdefiniowaną w programie, określającą rozmiary tablic A i B.

*Kod źródłowy funkcji MergeSort, realizującej algorytm sortowania przez scalanie:*

```
63
64 void MergeSort(int A[], int p, int k)
65 {
66     if(p < k)
67     {
68         int s = (p + k)/2;
69         MergeSort(A, p, s);
70         MergeSort(A, s+1, k);
71         Merge(A, p, s, k);
72     }
73 }
74
```

## Ćwiczenie 1

Napisz i uruchom program porządkujący tablice liczb metoda sortowania przez scalanie. Pamiętaj, żeby zastosować funkcje losująca i funkcje wypisująca tablice liczb. W funkcji main wywołaj funkcje MergeSort dla wartości parametru p równej 0 i wartości parametru k równej N - 1 (od pierwszego do ostatniego elementu tablicy).

## Podpowiedzi do ćwiczenia

*Kod źródłowy funkcji losującej i wypisującej tablicę liczb,*

```
6  #include <cstdlib>
7  #include <ctime>
8
9  using namespace std;
10
11 const int N = 20;
12
13 void Losuj(int A[])
14 {
15     for(int i = 0; i < N; i++)
16         A[i] = rand() % 100;
17 }
18
19 void Wypisz(int A[])
20 {
21     for(int i = 0; i < N; i++)
22         cout << A[i] << " ";
23     cout << endl;
24 }
25
```

*Kod źródłowy funkcji main programu sortującego liczby z wykorzystaniem funkcji MergeSort.*

```
74
75 int main()
76 {
77     int A[N];
78     srand(time(NULL));
79     Losuj(A);
80     Wypisz(A);
81     MergeSort(A, 0, N-1);
82     Wypisz(A);
83     return 0;
84 }
85
```

## Podsumowanie najważniejszych algorytmów sortowania.

<i>Algorytm</i>	<i>Złożoność obliczeniowa przypadek ogólny</i>	<i>Złożoność obliczeniowa przypadek optymistyczny</i>	<i>Złożoność obliczeniowa przypadek pesymistyczny</i>	<i>Złożoność pamięciowa nie licząc wejścia</i>	<i>Zastosowanie praktyczne</i>
Bubble sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	brak
Quick sort	$O(n \cdot \log n)$	$O(n \cdot \log n)$	$O(n^2)$	$O(\log n)$	ogólny przypadek
Merge sort	$O(n \cdot \log n)$	$O(n \cdot \log n)$	$O(n \cdot \log n)$	$O(n)$	sortowanie rozproszone

### 3 Szukamy różnych podciągów.

Czasami zachodzi potrzeba znalezienia fragmentu danych spełniającego pewne warunki. Na przykład trener zapisuje wyniki zawodnika osiągnięte podczas treningów, aby się dowiedzieć, kiedy rezultaty są coraz lepsze, a kiedy następuje spadek formy. Dużą popularnością cieszą się aplikacje, za pomocą których możemy monitorować swoją aktywność (sen, ruch, wysiłek fizyczny itp.), a potem np. sprawdzać, jaki był najdłuższy okres regularnych treningów albo kiedy spaliśmy najkrócej. W tym temacie zajmiemy się poszukiwaniem danych spełniających określone warunki.

#### 3.1 Szukamy długości najdłuższego spójnego podciągu niemalejącego.

Podciąg to wybrane elementy ciągu wyjściowego zachowujące kolejność występowania. Na przykład liczby 3, 2, 5, 7 tworzą podciąg ciągu 3, 1, 2, 5, 7, 4. Jeśli elementy podciągu występują w ciągu wyjściowym obok siebie, to taki podciąg nazywamy podciągiem spójnym. Na przykład ciąg 1, 2, 5 jest podciągiem spójnym ciągu 3, 1, 2, 5, 7, 4.

W podciągu niemalejącym każdy kolejny element jest większy lub równy poprzedniemu. Na przykład w ciągu 3, 1, 2, 5, 7, 4 możemy wyróżnić następujące spójne podciągi niemalejące:

- 1,2
- 1,2,5
- 1,2,5,7
- 2,5
- 2,9, 7
- 5,7

W powyższym przykładzie najdłuższy spójny podciąg niemalejący tworzą liczby: 1, 2, 5, 7. Jego długość jest równa 4.



Długość najdłuższego podciągu niemalejącego będzie równa jeden, gdy ciąg wyjściowy jest malejący.

Podciągi **jednoelementowe** potraktujemy jako podciągi **niemalejące**. Może się więc zdarzyć, że szukana długość będzie równa jeden. Może być też równa długości ciągu wyjściowego.

Algorytm poszukiwania długości najdłuższego spójnego podciągu niemalejącego jest podobny do znajdowania maksimum w tablicy. Długość aktualnie badanego fragmentu ciągu będziemy pamiętać w zmiennej *akt\_dl*, a największa dotychczas znaleziona długość podciągu spójnego — w zmiennej *maks\_dl*. Ponieważ każdy element ciągu tworzy podciąg jednoelementowy, początkowe wartości zmiennych *akt\_dl* i *maks\_dl* będą równe 1.

Pętla główna przegląda kolejne elementy tablicy, zaczynając od drugiego. Jeśli dany element tablicy jest większy lub równy poprzedniemu, to powiększamy wartość zmiennej *akt\_dl* o 1.

Ponadto sprawdzamy, czy długość badanego podciągu jest większa od dotychczas znalezionej największej długości podciągu. Jeśli jest większa, zmieniamy wartość zmiennej *maks\_dl*. Gdy przeglądany element tablicy jest mniejszy od poprzedniego, staje się on początkiem następnego podciągu, a więc wartość zmiennej *akt\_dl* musimy ustawić ponownie na 1.

*Specyfikacje problemu znajdowania długości najdłuższego spójnego podciągu niemalejącego przykładowy zapis w pseudokodzie.*

**Dane:**  $A[0..n-1]$  — tablica  $n$  liczb całkowitych.

**Wynik:**  $maks\_dl$  — długość najdłuższego spójnego podciągu niemalejącego w tablicy  $A$ .

$maks\_dl \leftarrow 1$

$akt\_dl \leftarrow 1$

**dla**  $i \leftarrow 1, \dots, n - 1$  **wykonuj**

**jeśli**  $A[i] \geq A[i-1]$  **to**

$akt\_dl \leftarrow akt\_dl + 1$

**jeśli**  $akt\_dl > maks\_dl$  **to**  $maks\_dl \leftarrow akt\_dl$

**w przeciwnym przypadku**  $akt\_dl \leftarrow 1$

## Ćwiczenie 1

Napisz program znajdujący długość najdłuższego spójnego podciągu niemalejącego. Wykorzystaj w nim zdefiniowaną wcześniej funkcję DNSPNM.

*Fragment kodu źródłowego programu poszukującego długości najdłuższego spójnego podciągu niemalejącego definicja funkcji znajdującej te długość*

```
33
34 int DNSPNM(int A[])
35 {
36     int maks_dl = 1, akt_dl = 1, i;
37     for(i = 1; i < N; i++)
38     {
39         if(A[i] >= A[i-1])
40         {
41             akt_dl++;
42             if(akt_dl > maks_dl)
43                 maks_dl = akt_dl;
44         }
45         else
46             akt_dl = 1;
47     }
48
49     return maks_dl;
50 }
51
```

## 3.2 Szukamy maksymalnej sumy podciągu spójnego.

Kolejnym problemem, którym się zajmiemy, będzie znalezienie największej sumy wyrazów wśród podciągów spójnych danego ciągu. Gdyby w tablicy występowały tylko liczby dodatnie, rozwiązanie byłoby oczywiste - największą sumę wyrazów ma podciąg taki sam jak ciąg wyjściowy. Problem staje się trudniejszy, gdy w tablicy występują także liczby ujemne.

Weźmy pod uwagę na przykład ciąg liczb:

23, - 49, 6, 23, 24, - 42, 40, 4, - 48, 39

Szukana suma wynosi 55. Jest ona suma wyrazów podciągu złożonego z liczb:  
6, 23, 24, -42, 40, 4.

*Specyfikacja problemu znajdowania maksymalnej sumy podciągu i zapis w pseudokodzie:*

**Dane:**  $A[0..n-1]$  - tablica  $n$  liczb całkowitych, co najmniej jedna liczba w tablicy jest nieujemna.

**Wynik:** maks\_suma - maksymalna suma elementów podciągu spójnego w tablicy  $A$ .

Najprostszym rozwiązaniem wydaje się liczenie sum wyrazów we wszystkich podciągach spójnych. Jeśli aktualnie obliczona suma będzie większa od dotychczas znalezionej, zmieniamy wartość maksymalnej sumy. Powinniśmy zliczać sumę wszystkich podciągów rozpoczynających się od pierwszego elementu tablicy, następnie wszystkich podciągów rozpoczynających się od drugiego elementu itd. Wymaga to użycia potrójnie zagnieżdżonej pętli. Aktualnie liczoną sumę podciągu zapamiętamy w zmiennej akt\_suma, a największą znaną sumę w zmiennej maks\_suma.



Algorytm zapisany w pseudokodzie może wyglądać:

```
maks_suma ← 0
dla i ← 0, ... n - 1 wykonuj
    dla j ← i, ..., n - 1 wykonuj
        akt_suma ← 0
        // liczymy sumę podciągu od i do j
        dla k ← i, ..., j wykonuj
            akt_suma ← akt_suma + A [k]
        jeśli akt_suma > maks_suma to
            maks_suma ← akt_suma
```

### Ćwiczenie1

Napisz i przetestuj program znajdujący największą sumę podciągu spójnego z wykorzystaniem powyższych funkcji MSPS i Losuj.

W programie realizującym podany algorytm potrzebna jest funkcja losująca tablice liczb, w której co najmniej jedna liczba będzie nieujemna. Poniższa funkcja losuje liczby całkowite z przedziału [-50; 49].

W linii 21 w losowym miejscu tablicy dodatkowo wstawiana jest losowa liczba nieujemna. Dzięki temu mamy pewność, że w tablicy nie znajdą się same liczby ujemne.

*Kod źródłowy funkcji realizującej algorytm MSPS , losującej tablicę liczb i ją wypisującą.*

```
13
14 void Losuj(int A[])
15 {
16     int i;
17     for(i = 0; i < N; i++)
18     {
19         A[i] = rand()%100 - 50;
20     }
21     A[rand()%N] = rand()%50;
22 }
23
24 void Wypisz(int A[])
25 {
26     int i;
27     for(i = 0; i < N; i++)
28     {
29         cout << A[i] << " ";
30     }
31     cout << endl;
32 }
33
```



```
33
34 int MSPS(int A[])
35 {
36     int maks_suma = 0, akt_suma, i, j;
37
38     for(i = 0; i < N; i++)
39     {
40         akt_suma = 0;
41         for(j = i; j < N; j++)
42         {
43             akt_suma += A[j];
44
45             if(akt_suma > maks_suma)
46                 maks_suma = akt_suma;
47         }
48     }
49 }
```

### 3.3 Szukamy podciągu spójnego o maksymalnej sumie.

Aby znaleźć podciąg o maksymalnej sumie, należy pamiętać indeks początku badanego podciągu oraz albo jego sumę, albo indeks końca. W celu wyznaczenia indeksu pierwszego elementu podciągu wystarczy zmodyfikować wybrany algorytm liczenia maksymalnej sumy podciągu.

*Liniowy algorytm znajdowania maksymalnej sumy, uzupełniony o wyznaczenie indeksu pierwszego elementu podciągu, zapisany w pseudokodzie:*

```
maks_suma ← 0
akt_suma ← 0
maks_pocz ← 0
akt_pocz ← 0
dla i ← 0, ... n - 1 wykonuj
    jeśli akt_suma + A[i] > 0 to
        akt_suma ← akt_suma + A[i]
        jeśli akt_suma > maks_suma to
            maks_suma ← akt_suma
            maks_pocz ← akt_pocz
    w przeciwnym przypadku
        akt_suma ← 0
        akt_pocz ← i + 1
```

Funkcja realizująca powyższy algorytm musi zwrócić dwie wartości: znaleziona sumę oraz indeks pierwszego elementu podciągu. Dlatego prześlemy obie te wartości przez referencje. Funkcja nie będzie miała wartości, będzie więc typu void.

*Fragment kodu źródłowego programu znajdującego podciąg spójny o maksymalnej sumie - funkcja znajdująca pierwszy element oraz maksymalną sumę podciągu.*

```
34
35 void PSMS(int A[], int &maks_suma, int &maks_pocz)
36 {
37     maks_suma = 0;
38     maks_pocz = 0;
39     int akt_suma = 0, akt_pocz = 0, i;
40
41     for(i = 0; i < N; i++)
42     {
43         if(akt_suma + A[i] > 0)
44         {
45             akt_suma += A[i];
46             if(akt_suma > maks_suma)
47             {
48                 maks_suma = akt_suma;
49                 maks_pocz = akt_pocz;
50             }
51         }
52         else
53         {
54             akt_suma = 0;
55             akt_pocz = i + 1;
56         }
57     }
58 }
59
```

Zmieniając wartość dotychczas znalezionej największej sumy (linia 48), zmieniamy także wartość indeksu pierwszego elementu podciągu (linia 49). W przypadku, gdy dodanie kolejnego elementu tablicy da sumę niedodatnią, badany jest kolejny podciąg rozpoczynający się od następnego elementu tablicy (jego indeksem jest wartość  $i+1$ , linia 55).

Program wykorzystujący powyższa funkcje podczas wypisywania podciągu musi zliczać na nowo sumę jego wyrazów: od pierwszego wyrazu podciągu do momentu, gdy liczona suma będzie równa maksymalnej sumie.

*Fragment kodu źródłowego programu znajdującego podciąg o maksymalnej sumie - funkcja main*

```
60 int main()
61 {
62     int suma, pocz, tmp = 0, i;
63     int A[N];
64
65     srand(time(NULL));
66
67     Losuj(A);
68     Wypisz(A);
69     PSMS(A, suma, pocz);
70
71     cout<<"Maksymalna suma podciagu spojnego: " << suma << endl;
72
73     cout<<"Podciag spojny o maksymalnej sumie: " << endl;
74
75     i = pocz;
76     while(tmp != suma)
77     {
78         cout << A[i] << " ";
79         tmp += A[i];
80         i++;
81     }
82
83     return 0;
84 }
85
```



### Ćwiczenie 1

Napisz i przetestuj program wypisujący podciąg spójny o maksymalnej sumie wykorzystując powyższe funkcje..



# Tworzenie stron internetowych.

## 4 HTML

Większość witryn internetowych tworzonych jest obecnie w oparciu o system zarządzania treścią (CMS). Pozwala on zarządzać treścią witryny bez znajomości HTML i CSS. Znajomość języka znaczników, choćby w podstawowym zakresie, może być jednak przydatna w codziennej pracy redaktora treści. Wybrane formatowanie tekstu, elementów witryny lub modyfikacje CMS mogą bowiem wymagać ingerencji w kod źródłowy.

Język HTML (ang. *Hypertext Markup Language*) to język, który za pomocą tzw. **znaczników** lub inaczej **tagów** opisuje strukturę strony i jej zawartość. Sama znajomość tagów nie wystarcza, żeby posługiwać się tym językiem należy rozumieć jego składnię.

Strona internetowa to plik tekstowy, w którym za pomocą specjalnego języka opisano elementy strony i jej wygląd. Może mieć rozszerzenie html lub htm. Serwisy internetowe składają się z wielu powiązanych ze sobą stron internetowych.

Do pisania kodu będziemy wykorzystywać darmowy program Notepad++, ułatwiający pracę m.in. dzięki kolorowaniu składni języka HTML.

W przeglądarce internetowej będziemy uruchamiać i testować nasze strony.

Plik będący stroną startową (czyli główną) musi nazywać się **index.html** lub **index.htm**.

**UWAGA!** Bardzo ważne: w nazwach plików lepiej nie używać:

- wielkich liter
- znaków specjalnych, np.: \ / : \* ? " < > |
- spacji (jeśli musisz, w zamian używaj podkreślnika "\_")
- polskich liter (ą, ć, ę, ł, ń... itd.)

### Ćwiczenie 1

Utwórz katalog Strona\_WWW. W edytorze Notepad++ utwórz plik i zapisz go pod nazwą index.html.

## 4.1 Podstawowe znaczniki HTML – szkielet strony.

### *Budowa znacznika w języku HTML.*

Poza pewnymi wyjątkami znaczniki występują parami:

<a> znacznik otwierający,

</a> znacznik zamykający.

Powyższy znacznik służy do tworzenia hiperłączy (linków) do plików lub innych stron internetowych.

Możliwości znaczników możemy rozszerzać za pomocą atrybutów np.:

<a href="https://lo3.edu.pl">> kliknij tu </a>

Pozwala on przejście do strony naszego liceum.

Prosty dokument hipertekstowy (HTML5) ma następującą strukturę:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Tytuł strony WWW</title>
</head>
<body>
<h1>Nagłówek</h1>
<p>Paragraf</p>
</body>
</html>
```

Deklaracja <!DOCTYPE html> definiuje dokument jako HTML5.

Element <html> jest głównym elementem strony HTML.

Cała treść która znajduje się pomiędzy <head> oraz </head> (w szablonie powyżej) jest nazywana *nagłówkiem dokumentu* (treść nagłówkowa), natomiast wszystko pomiędzy <body> oraz </body> stanowi tzw. *ciało dokumentu* (treść właściwa).

Część HEAD odpowiada za opis strony.

Poniżej kilka przykładów zawartości HEAD oraz wyjaśnienie roli jaką pełnią.

### *O czym nie należy zapominać pisząc stronę WWW po polsku?*

```
<head>
  <meta charset="utf-8">
  .....
</head>
```

Polecenie jest deklaracją strony kodowej, czyli sposobu w jaki będą zapisane w formie cyfrowej znaki na naszej stronie WWW.

Nie należy pomijać tego wpisu i powinna to być koniecznie deklaracja kodowania UTF, a nie WINDOWS, gdyż wtedy mogą wystąpić problemy z wyświetlaniem polskich znaków (nawet jeśli na Twoim komputerze wszystko jest w porządku)!

### *W jaki sposób zachęcić internautów aby odwiedzili Twoją stronę? Jak wstawić opis, który wyświetli się w wyszukiwarkach (Google)?*

```
<head>
  <meta name="description" content="Tu podaj opis twojej strony">
</head>
```

Pozwala ono opisać co znajduje się na Twojej stronie. Z informacji tej korzystają wyszukiwarki sieciowe, dlatego staraj się tutaj wpisać tekst, który jak najlepiej opisze zawartość strony i zachęci do jej odwiedzenia. Ciekawy, ale niezbyt długi, opis może zachęcić internautów do obejrzenia Twojej strony!

### *W jaki sposób podwyższyć pozycję strony WWW w wyszukiwarkach (Google)?*

```
<head>
  <meta name="keywords" content="wyraz1, wyraz2, wyraz3...">
</head>
```

gdzie "wyraz1, wyraz2, wyraz3..." oznaczają wyrazy, które należy rozdzielać przecinkami. Można oczywiście podać więcej niż trzy wyrazy (w miejsce kropek).

Polecenie to należy wstawić między znacznikami: <head> oraz </head>. Pozwala Ci ono podać wyrazy kluczowe, z których korzystają wyszukiwarki sieciowe. Dlatego staraj się tutaj wpisać wyrazy, które jak najlepiej opiszą zawartość Twojej strony. Dobrze dobrane wyrazy kluczowe, pomogą wyszukiwarkom odnaleźć Twoją stronę!

### Gdzie wpisać autora strony WWW?

```
<head>  
  <meta name="author" content="Tu wpisz swoje imię i nazwisko">  
  <meta name="creator" content="Tu wpisz nazwę organizacji">  
  
  <meta name="publisher" content="Tu wpisz nazwę wydawcy">  
</head>
```

Pierwsze polecenie ("author") pozwala podać informację o fizycznej osobie będącej autorem strony, natomiast drugie ("creator") i trzecie ("publisher") - odpowiednio: informację o nazwie organizacji i wydawcy strony. *Wydawca* to firma lub instytucja, której główną działalnością jest publikowanie treści, podczas gdy *organizacja* może się zajmować również czymś innym.

### Jak wstawić informację o edytorze użytym przy tworzeniu strony WWW?

```
<head>  
  <meta name="generator" content="nazwa edytora">  
</head>
```

gdzie jako "**nazwa edytora**" można wpisać nazwę programu, przy użyciu którego została stworzona strona.

Pozwala określić nazwę edytora HTML, który został wykorzystany do utworzenia dokumentu.

## W jaki sposób przeskalować rozmiar strony WWW? Jak przygotować stronę RWD (Responsive Web Design)?

```
<head>  
  <meta name="viewport" content="parametr1=wartość1, para-  
metr2=wartość2...">  
</head>
```

gdzie jako "**parametr1=wartość1, parametr2=wartość2...**" należy podać rozdzieloną przecinkami listę parametrów wraz z przypisanymi im wartościami:

### **width**

Określa początkową szerokość obszaru wyświetlania przeglądarki. Może być określona bezpośrednio w pikselach albo za pomocą specjalnego słowa kluczowego `device-width`, które oznacza pełną szerokość obszaru wyświetlania.

### **height**

Określa początkową wysokość obszaru wyświetlania przeglądarki. Może być określona bezpośrednio w pikselach albo za pomocą specjalnego słowa kluczowego `device-height`, które oznacza pełną wysokość obszaru wyświetlania.

### **initial-scale**

Wartość od 0.0 do 10.0 (należy posługiwać się kropką dziesiętną, a nie przecinkiem!) określająca początkowy stopień przeskalowania strony. Im większa wartość tym bardziej będzie powiększona zawartość strony.

### **minimum-scale**

Wartość od 0.0 do 10.0 określająca minimalny stopień, do którego użytkownik będzie mógł przeskalować stronę. Jest to przydatne w przypadku, gdy nasza strona nie wygląda zbyt dobrze, kiedy jest za bardzo pomniejszona. Pamiętajmy jednak, że stanowi to ingerencję w preferencje użytkowników i dlatego przeglądarki mogą blokować to polecenie.

### **maximum-scale**

Wartość od 0.0 do 10.0 określająca maksymalny stopień, do którego użytkownik będzie mógł przeskalować stronę. Przeglądarki mogą blokować to polecenie.

### **user-scalable**

Może przyjąć następujące wartości:

- **yes** - domyślnie
- **no** - możliwość skalowania strony zostanie zablokowana (przeglądarki mogą ignorować to polecenie z uwagi na preferencje użytkownika)

Polecenie to jest szczególnie przydatne na urządzeniach mobilnych takich jak smartfony czy tablety. Ponieważ ekran przeglądarek mobilnych zwykle jest dość małych rozmiarów, strona może być odpowiednio powiększana, tak aby jej zawartość była dobrze czytelna.



W pewnym zakresie możemy sterować tym zachowaniem za pomocą powyższego polecenia. W szczególności możemy zaprojektować stronę z użyciem techniki *Responsive Web Design* (w skrócie: *RWD*), tak aby automatycznie dopasowywała się do dostępnej szerokości ekranu przeglądarki. W takim przypadku w nagłówku strony powinniśmy wstawić następujące polecenie:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

## Ćwiczenie 2

1. Otwórz w edytorze HTML stronę (plik `index.html`) stworzoną w ramach ćwiczeń z poprzedniego rozdziału.
2. Dodaj informację o języku, w którym została wprowadzona treść na Twojej stronie.
3. Podaj autora strony.

## 4.2 Formatowanie treści strony za pomocą znaczników HTML.

### Akapit <p>

Akapit jest to część tekstu objęta znacznikami: <p>...</p>. Następujące bezpośrednio po sobie akapity są oddzielone przerwą w tekście (pustą linią). Dzięki umieszczaniu w kolejnych akapitach treści nieco różniącej się tematycznie, strona stanie się bardziej estetyczna oraz czytelniejsza.

Akapit (paragraf) jest bardzo ważny w składni HTML, ponieważ pozwala w określony sposób sformatować tekst na ekranie (ułożyć go w podany sposób). Robi się to podając *atrybuty znacznika*. Atrybut wpisuje się zawsze wewnątrz znacznika otwierającego - bezpośrednio po jego nazwie (oddzielony od niej spacją), a przed znakiem zamknięcia nawiasu ostrego, czyli przed ">". Każdy znacznik ma ściśle określone atrybuty, które obsługuje. W przypadku akapitu można zastosować m.in. następujące:

#### 1. Wyrównanie tekstu do lewej strony (domyślnie)

```
<p style="text-align: left">Treść akapitu</p>
```

lub po prostu

```
<p>Treść akapitu</p>
```

#### 2. Wyrównanie tekstu do prawej

```
<p style="text-align: right">Treść akapitu</p>
```

#### 3. Wyśrodkowanie tekstu

```
<p style="text-align: center">Treść akapitu</p>
```

#### 4. Justowanie tekstu (wyrównanie do obu marginesów jednocześnie)

```
<p style="text-align: justify">Treść akapitu</p>
```

## Tytuł <h1, h2, h3, h4, h5, h6>

Aby nadać tytuł (nagłówek) jakiejś części tekstu (rozdziałowi), możesz użyć tej komendy. Istnieje sześć rzędów tytułów. Najwyższym rzędem jest rząd pierwszy: <h1>...</h1>, a najniższym - szósty: <h6>...</h6>. Tytuł wyższego rzędu jest pisany większą czcionką.

## Blok <div>

Polecenie to wydziela większy blok tekstu. W odróżnieniu od akapitu, blok może zawierać wewnątrz siebie inne elementy wyświetlane w bloku. Kolejne bloki są oddzielane od siebie znakami nowej linii, ale nie są dodawane linijki przerwy (aby je dodać, należy zastosować znacznik <p>...</p> albo <br>).

## Koniec linii <br>

Jest to bardzo przydatny znacznik. Używamy go, gdy chcemy natychmiast zakończyć linię i przejść do następnej (wszystkie normalne znaki końca linii są ignorowane przez przeglądarkę).

## Pogrubienie <b>

Znacznik ten pozwala pogrubić (wytluszczyć) część tekstu (ang. "bold"). Oznacza on treść, na którą należy zwrócić uwagę z powodów czysto użytkowych. Nie nadaje dodatkowej ważności treści ani nie stawia akcentu wypowiedzi. Przydatne przy oznaczaniu słów kluczowych w tekście albo wprowadzenia na początku artykułu.

## Pochylenie <i>

Pozwala napisać tekst pismem pochylonym, czyli kursywą (ang. "italic"). Oznacza np. termin techniczny, idiom z innego języka albo fragment wtrąconego tekstu.

## Podkreślenie <u>

Pozwala podkreślić fragment tekstu (ang. "underline"). Oznacza nieartykułowany tekst albo oznaczenie błędu ortograficznego.

## Indeks górny i dolny <sup, sub>





## 1. Indeks górny

```
<sup>...</sup>
```

## 2. Indeks dolny

```
<sub>...</sub>
```

Umożliwia wprowadzenie indeksów (górných lub dolnych) przy cyfrach i literach.

### Pozioma linia <hr>

```
<hr>
```

Polecenie to pozwala narysować na ekranie poziomą linię. Może ona np. rozdzielać kolejne rozdziały, które różnią się tematycznie, przez co strona staje się bardziej czytelna i przejrzysta.



## 4.3 Listy, odsyłacze i grafika na stronach WWW.

### Listy <ul, ol, li>

#### Wypunktowanie - lista nieuporządkowana <ul, li>

```
<ul>
<li>Punkt pierwszy</li>
<li>Punkt drugi</li>
<li>Punkt trzeci</li>
</ul>
```

#### Wykaz numerowany - lista uporządkowana <ol, li>

```
<ol>
<li>Punkt pierwszy</li>
<li>Punkt drugi</li>
<li>Punkt trzeci</li>
</ol>
```

#### Wykaz numerowany - różne typy

```
<ol type="rodzaj numeracji">
<li>Punkt pierwszy</li>
<li>Punkt drugi</li>
<li>Punkt trzeci</li>
</ol>
```

gdzie jako **"rodzaj numeracji"** należy podać:

- "1" (domyślny) - numeracja według liczb arabskich
- "I" - według dużych liczb rzymskich
- "i" - według małych liczb rzymskich
- "a" - według małych liter
- "A" - według dużych liter



## Multimedia

### Obrazek <img>

```

```

gdzie jako "ścieżka dostępu" należy podać lokalizację na dysku, gdzie znajduje się żądany obrazek. Natomiast w miejsce "Tekst alternatywny" wpisuje się krótką informację, która pojawi się w przypadku, kiedy obrazek nie zostanie wyświetlony (np. jeśli użytkownik wyłączy wyświetlanie grafiki w swojej przeglądarce internetowej). Ponadto w MSIE może zostać wyświetlona po wskazaniu obrazka myszką.

Informacja alternatywna jest przydatna w przeglądarkach, które nie wyświetlają grafiki lub kiedy użytkownik wyłączył jej wyświetlanie albo dla osób niewidomych, które korzystają ze specjalnych syntezatorów mowy.

### Obrazek o określonych rozmiarach <img width height>

```

```

lub

```

```

gdzie jako "x" oraz "y" należy wpisać odpowiednio: długość i wysokość obrazka w pikselach. Natomiast jako "x%" oraz "y%" należy wpisać odpowiednio: długość i wysokość obrazka w procentach ekranu.

### Osadzenie pliku <embed>

```
<embed src="ścieżka dostępu do pliku" width="x" height="y">
```

gdzie jako "ścieżka dostępu do pliku" należy podać lokalizację na dysku, gdzie znajduje się żądany plik multimedialny.

Natomiast "x" oznacza szerokość w pikselach, a "y" wysokość wyświetlanego obrazu wtyczki (ang. plug-in), czyli okna na stronie w którym odtwarzany będzie plik. Jeśli nie podamy rozmiarów obrazu wtyczki, może on przyjąć wielkość, która nie pasuje do wymiarów pliku! Rozmiary zależą również od posiadanej przez użytkownika wtyczki.

Polecenie <embed> jest przydatne jeśli chcemy wstawić na stronę jeden z plików multimedialnych:

- \*.wav - plik dźwiękowy typu "wav"
  - \*.mid - plik dźwiękowy typu "midi"
  - \*.avi - plik typu "avi"
  - \*.ra - plik Real Audio Player
  - \*.mp3 - plik dźwiękowy typu "mp3" (MPEG Layer-3)
  - \*.mpeg - plik typu "mpeg"
  - \*.mov - plik typu "mov"
  - \*.asf - plik typu "asf"
- i inne

## Odsyłacze

Odsyłacze - inaczej hiperłącza, odnośniki hipertekstowe, linki - stanowią całą istotę Internetu. Bez nich byłby on jedynie wielką siecią do pobierania "suchych" danych, a dokumenty znajdujące się w niej, nie byłyby w żaden sposób ze sobą powiązane. Praktycznie na każdej stronie WWW spotyka się odsyłacze. Najczęściej stanowi je specjalnie wyróżniony krótki tekst (lub obrazek), po kliknięciu którego, następuje przeniesienie do innej strony.

Przy czym strona taka może wchodzić w skład tego samego serwisu, ale równie dobrze może znajdować się na drugim końcu świata. Przeglądarki internetowe wyświetlają odsy-

łącze najczęściej w innym kolorze oraz podkreślają je (można to oczywiście zmienić) – w ten sposób są one lepiej widoczne i odróżniają się od zwykłego tekstu.

### Do podstrony <a href>

Serwisy internetowe składają się zwykle z podstron. Każda podstrona jest osobnym plikiem HTML i zawiera treść, która dość znacznie różni się tematycznie od pozostałych. Taka organizacja ułatwia użytkownikowi nawigację oraz zapobiega niepotrzebnemu wczytywaniu całego serwisu od razu (co trwałoby prawdopodobnie bardzo długo).

Aby umożliwić w nich nawigację, wstawia się poniższe polecenie, odnoszące się do wszystkich podstron serwisu. Jest to pomocne przy tworzeniu spisu treści, chociaż hiperłącza mogą znajdować się również w zwykłym tekście, odsyłając użytkownika do miejsca, gdzie dany temat jest szerzej opisany. Odsyłacz taki może odnosić się do innej podstrony, ale tylko w obrębie tego samego dysku (ten sam serwis internetowy).

```
<a href="względna ścieżka dostępu">opis</a>
```

gdzie jako "względna ścieżka dostępu" należy podać lokalizację na dysku, gdzie znajduje się podstrona, do której chcemy się przenieść.

W miejsce wyrazu "opis" wpisujemy krótki tekst, który pojawi się na ekranie i po kliknięciu którego nastąpi uruchomienie odsyłacza (ścieżka dostępu pojawia się jedynie na pasku statusu przeglądarki).

### Do adresu internetowego <a href http https> (w obrębie całego Internetu)

#### Połączenie niezabezpieczone

```
<a href="http://adres strony">opis</a>
```

#### Bezpieczne połączenie SSL

```
<a href="https://adres strony">opis</a>
```

Protokół SSL (Secure Sockets Layer - Bezpieczna Warstwa Gniazdek) został opracowany przez firmę Netscape Communications Corporation. Jest używany do przesyłania poufnych danych. Teoretycznie za jego pomocą można nawiązać połączenie ze zwykłą stroną HTML - analogicznie jak w przypadku niezabezpieczonego odsyłacza do adresu internetowego. Jednak aby to zrealizować, serwer musi obsługiwać tego typu połączenia, a dodatkowo posiadać własny certyfikat, dzięki któremu użytkownik może być pewny, że łączy się ze stroną, której żądał. Ponieważ dane są szyfrowane, więc nawet jeśli osoba niepowołana przechwyci je po drodze podczas przesyłania, nie będzie potrafiła ich odczytać. Protokół SSL jest powszechnie stosowany przez sklepy i banki internetowe.

### Odsyłacz obrazkowy <a href, img, >

Na większości stron internetowych można spotkać "aktywne" obrazki, symulujące przyciski. Po kliknięciu, zachowują się one jak zwykły odsyłacz (w istocie są one odsyłaczami). Wprowadzenie takich przycisków na stronę jest prostsze niż myślisz - wystarczy pamiętać, że wewnątrz znacznika odsyłacza (pomiędzy <a href="..."> a </a>) można umieszczać nie tylko tekst.

### Podstawowy odsyłacz obrazkowy <a href, img>

```
<a href="adres"></a>
```

gdzie jako "ścieżka dostępu" należy podać lokalizację na dysku, gdzie znajduje się żądany obrazek.

Zasady wpisywania adresu są analogiczne jak w przypadku odsyłaczy do: podstrony, etykiety, adresu internetowego, poczty e-mail czy dowolnych innych odsyłaczy (w zależności od wybranego typu odsyłacza).

Odsyłacz ten zostanie uruchomiony, gdy klikniemy myszką obrazek, do którego podajemy ścieżkę dostępu. Dzięki niemu możemy stworzyć np. efektowne przyciski odsyłaczowe w menu strony.

Obrazki przycisków najlepiej zapisywać w formacie GIF, większe zdjęcia - JPG. Jeśli nie masz zacięcia artystycznego, nie musisz samodzielnie rysować wszystkich grafik. W Internecie na pewno znajdziesz wiele stron, gdzie możesz darmowo pobrać gotowe przyciski.

Tekst alternatywny (atrybut alt="..." dla znacznika <img>) może pojawić się w "dymku" narzędziowym po wskazaniu obrazka myszką (Microsoft Internet Explorer). Jest to jednak przede wszystkim informacja dla przeglądarek tekstowych, które nie wyświetlają grafiki. Dzięki temu również w takich przeglądarkach możliwe będzie używanie odsyłacza obrazkowego, chociaż grafika nie zostanie wyświetlona.

## 5 CSS

### 5.1 Podstawy CSS

- I. Style stały się już praktycznie podstawowym narzędziem formatującym. Jeśli poważnie myślisz o zajęciu się projektowaniem stron WWW, koniecznie musisz je poznać.
- II. Przeszarzałe atrybuty i znaczniki, znajdujące się bezpośrednio w składni HTML, które dotyczą formatowania (zobacz: Elementy zdeprecjonowane), będą stopniowo wycofywane przez producentów przeglądarek internetowych, na rzecz rekomendowanych analogicznych deklaracji CSS. Widać już nawet różnice pomiędzy MSIE 4 a MSIE 5!
- III. Dokumenty pisane z wykorzystaniem arkuszy stylów są zwykle bardziej przejrzyste i krótsze.
- IV. Style pozwalają w łatwy sposób zarządzać całą serią dokumentów, poprzez stosowanie zewnętrznych arkuszy stylów. Dzięki temu w łatwy i wygodny sposób, można dokonać modyfikacji rodzaju formatowania jednocześnie we wszystkich dokumentach, zmieniając dane tylko w jednym pliku!
- V. Dzięki możliwości stosowania klas selektorów, znacznie oszczędzamy sobie pisanie. W jednym miejscu określamy wszystkie atrybuty formatowania (których może być bardzo dużo), odnoszące się do wielu elementów, które mają wyglądać tak samo. Bezpośrednio przy elemencie wystarczy podać tylko nazwę klasy i nie musimy już wypisywać "litanii" poleceń. Znacznie przyspiesza to późniejszą modyfikację strony, ponieważ zmian dokonujemy tylko w jednym miejscu, a wpływają one na wiele elementów jednocześnie.
- VI. Możliwość stosowania różnorodnych jednostek oraz sposobów definiowania kolorów.
- VII. Style dają autorowi możliwości, które do tej pory były praktycznie niemożliwe do osiągnięcia:



- a) Różne wartości pogrubienia czcionki (9 rodzajów)
- b) Dodatkowe możliwości formatowania tekstu

### Wygląda typowego arkusza CSS?

Arkusz stylów jest zwykłym dokumentem tekstowym, więc do jego stworzenia możesz użyć Twojego ulubionego edytora HTML albo, jeśli wolisz, dedykowanego edytora CSS. W pliku tym wpisuje się listę tzw. reguł stylów, za pomocą których możliwe jest określanie wyglądu elementów na stronie. Każda reguła stylów składa się zawsze z dwóch następujących bezpośrednio po sobie części:

**Selektora** - czyli elementu (znacznika) na stronie, któremu chcemy zmienić wygląd

**Deklaracji** ujętej w nawiasy klamrowe, w której określa się, w jaki sposób chcemy zmienić wygląd

```
selektor1 {  
    Tu wpisuje się deklaracje stylów  
}  
  
selektor2 {  
    Tu wpisuje się deklaracje stylów  
}  
  
selektor3 {  
    Tu wpisuje się deklaracje stylów  
}
```

## 5.2 Źródła stylów

### Wstawienie arkusza stylów

**Kaskadowe Arkusze Stylów CSS** nie mogą funkcjonować samodzielnie, ponieważ definiują jedynie sposób formatowania (wygląd) elementów dokumentu, ale same ich nie tworzą. Elementy muszą zostać wstawione do dokumentu w postaci struktury znaczników np. za pomocą języka XHTML lub HTML. Dlatego aby poznać ogrom dodatkowych możliwości, jakie dają style CSS, konieczna jest wcześniejsza znajomość zasad języka HTML.

Zapamiętaj zatem prostą zasadę: za pomocą języka HTML wstawia się znaczniki do kodu źródłowego strony, a następnie dzięki CSS można nadać im potem określony sposób wyświetlania.

Sposoby wstawiania stylów do gotowych dokumentów są różne. Nie znaczy to, że jedne są lepsze od drugich. Każdy sposób jest przydatny w innych sytuacjach. Większość witryn stosuje jednocześnie wszystkie z przedstawionych metod osadzania CSS - w zależności od konkretnej potrzeby.

## Typowy sposób dołączania arkusza CSS

### Zewnętrzny arkusz stylów <link stylesheet> @charset

Aby rozpocząć przygodę z językiem CSS, wystarczy wstawić w nagłówku strony - tzn. w dowolnym miejscu pomiędzy znacznikami <head> oraz </head> - jedną dodatkową linię:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="description" content="Tu wpisz opis zawartości strony">
  <title>Tu wpisz tytuł strony</title>
  <link rel="stylesheet" href="style.css"></head>

<body>

  Tu wpisuje się treść strony
</body>
</html>
```

Dzięki znacznikowi <link> do dokumentu zostanie automatycznie dołączony arkusz stylów. W nim właśnie wpisuje się wszystkie polecenia zmieniające wygląd strony. Ważne jest tylko, aby ten plik posiadał rozszerzenie \*.css. W powyższym przykładzie jest to plik pod nazwą "style.css", który znajduje się w tym samym katalogu co dokument \*.html. Można go oczywiście zapisać w innej lokalizacji - wtedy zamiast "style.css" trzeba podać pełną ścieżkę dostępu (tworzy się ją w analogiczny sposób, jak przy wstawianiu obrazka).

Warto podkreślić, że jeśli Twoja strona WWW składa się z wielu podstron, na każdej z nich możesz dołączyć ten sam plik arkusza stylów. Dzięki temu zmiany wykonane tylko w jed-

nym miejscu - czyli w pliku arkusza stylów - wpłyną na wygląd od razu wszystkich podstron Twojego serwisu.

## Wewnętrzny arkusz stylów <style>

```
<head>
    (...)
<style>
selektor { cecha: wartość; cecha2: wartość2... }
selektor2 { cecha: wartość; cecha2: wartość2... }
    (...)
</style>
    (...)
</head>
```

gdzie w sposób wyróżniony zaznaczono elementy składowe wewnętrznego arkusza stylów. Natomiast w miejsce kropek (...) można wpisać dalsze polecenia.

Selektorem może być praktycznie dowolny znacznik, np. p (akapit), h1 (tytuł), td (komórka tabeli) i inne. To właśnie elementom, które znajdują się w kodzie źródłowym, pomiędzy tymi znacznikami, nadajemy atrybuty formatowania opisane w arkuszu.

Jako "cecha" (w deklaracji stylu - powyżej) należy wpisać o jakie konkretnie atrybuty formatowania nam chodzi (opisane w kolejnych rozdziałach).

Natomiast jako wyraz "wartość" wpisujemy dokładną wartość atrybutu.

Zwróć uwagę, że jednemu selektorowi możemy nadać kilka atrybutów (cech). Są one wtedy rozdzielone średnikami.

Wewnętrzny arkusz stylów wstawia się zawsze w części nagłówkowej dokumentu (pomiędzy znacznikami <head> a </head>).

Można go zastosować, gdy elementy które pragniemy poddać formatowaniu, występują wielokrotnie na stronie i wszystkim chcemy nadać takie same atrybuty (inne niż domyślne). Na przykład chcemy, aby wszystkie wykazy miały automatycznie kolor niebieski. Wystarczy wpisać odpowiednią deklarację stylów w arkuszu (w treści nagłówkowej) i nie trzeba już nic dopisywać przy samym elemencie.



## 5.3 Kaskadowość stylów

Kaskadowość stylów określa pierwszeństwo w oddziaływaniu na te same elementy strony stylów z różnych źródeł. Na przykład: na naszej stronie używamy zewnętrznego arkusza stylów, deklaracji stylów w nagłówku strony, a także stylów typu inline, przy czym dotyczą one formatowania dokładnie tego samego elementu (np. kroju czcionki). Co się stanie w takim przypadku? Czy nie powstaną konflikty? A może komputer się "zawiesi"? Nic z tych rzeczy! A to właśnie ze względu na kaskadowość. To od niej wzięły swoją nazwę style: CSS - (ang. Cascading Style Sheets) Kaskadowe Arkusze Stylów.

Jeśli w dokumencie znajduje się kilka źródeł stylów, pierwszeństwo mają te, które znajdują się "bliżej" formatowanego elementu. Oddziaływanie stylów z arkuszy zewnętrznych może być modyfikowane przez style zdefiniowane w nagłówku dokumentu, to z kolei może być zmieniane przez style zdefiniowane bezpośrednio w ciele dokumentu (inline). Zatem priorytet ważności stylów (pierwszeństwo) wyglądałby tak:

1. Styl lokalny (inline)
2. Rozciąganie stylu (SPAN)
3. Wydzielone bloki (DIV)
4. Wewnętrzny arkusz stylów
5. Import stylów do wewnętrznego arkusza
6. Zewnętrzny arkusz stylów
7. Import stylów do zewnętrznego arkusza
8. (Atrybuty prezentacyjne HTML -  
np. `align="..."`, `bgcolor="..."`, `color="..."`, `height="..."`, `width="..."` i inne)

Style o wyższym priorytecie ważności (na początku listy) mają pierwszeństwo w modyfikowaniu elementów dokumentu. Style niżej w hierarchii mogą zmienić formatowanie jakiegokolwiek elementu, tylko jeśli style o wyższym priorytecie nie dotyczą tego konkretnego elementu. Na szczególną uwagę zasługuje również fakt, że style CSS z dowolnego źródła zawsze są ważniejsze niż atrybuty prezentacyjne przypisane elementom bezpośrednio w

kodeksie HTML, mimo iż wydawać by się mogło, że powinna tutaj zachodzić zasada taka jak w przypadku stylu lokalnego.

Kaskadowość stylów pozwala zatem pracować z kilkoma źródłami stylów, bez obawy o wystąpienie konfliktów. Takie rozwiązanie jest bardzo wygodne, ponieważ możemy np. zdefiniować ogólny wygląd naszego serwisu w zewnętrznym arkuszu i dodatkowo zmienić nieco wybrane strony poprzez wewnętrzny arkusz, a jeśli zajdzie konieczność zmodyfikować pojedyncze elementy za pomocą stylu lokalnego.

**UWAGA!** Polecenie dołączenia zewnętrznego arkusza powinno znajdować się w dokumencie wcześniej niż wewnętrzny arkusz. Odwrotna kolejność złamie zasady kaskadowości!

## 5.4 Selektory

### Kolor tekstu {color}

Aby zmienić wygląd jakiegoś elementu na stronie, trzeba go najpierw wskazać. W języku CSS robi się to za pomocą tzw. *selektora*. W najprostszym przypadku jest to nazwa wybranego znacznika, który wcześniej wstawiliśmy do naszego dokumentu HTML. Wszystko co znajduje się wewnątrz tak wskazanego znacznika - czyli zarówno tekst, jak i inne znaczniki - otrzyma style podane w *deklaracji*.

```
selektor { color: kolor }
```

Selektorem może być dowolny znacznik, w którym można wpisywać tekst, np. p - akapit, h1 - tytuł czy td - komórka tabeli.

### Wyrównanie tekstu {text-align}

```
selektor { text-align: wyrównanie }
```

Selektorem może być dowolny znacznik wyświetlany w bloku, np. p - akapit, h1 - tytuł czy td - komórka tabeli.

Natomiast jako "wyrównanie" należy podać:

left - wyrównanie tekstu do lewego marginesu (domyślnie)

right - wyrównanie do prawego marginesu

center - do środka (wyrównanie)

justify - do obu marginesów jednocześnie (justowanie)

Polecenie pozwala wybrać jeden z możliwych sposobów wyrównania tekstu, czyli jego ułożenia na ekranie.

## Wcięcie w tekście {text-indent}

```
selektor { text-indent: wcięcie }
```

Selektorem może być każdy znacznik wyświetlany w bloku. Natomiast jako "wcięcie" należy podać konkretną wielkość wcięcia, używając jednostek długości.

Wcięcie w tekście oznacza dodatkowy odstęp pierwszej linijki w bloku od lewego marginesu (jak tabulator). Stosuje się je, gdy rozpoczynamy nowy akapit (na inny temat).

## Kolor tła {background-color}

```
selektor { background-color: kolor }
```

Selektorem może być praktycznie dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli. Natomiast jako "kolor" należy podać definicję koloru. Wpisanie "transparent" ustali tło przezroczyste.

Polecenie umożliwia wstawienie tła o określonym kolorze pod danym elementem (tekstem).

## Tło obrazkowe {background-image}

### Tło obrazkowe {background-image}

```
selektor { background-image: url(ścieżka dostępu do obrazka) }
```

Selektorem może być praktycznie dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli

Natomiast jako "ścieżka dostępu do obrazka" należy podać miejsce, gdzie znajduje się obrazek, który chcemy wstawić jako tło. Wpisanie "none" usunie obrazek.

*Ścieżkę dostępu należy konstruować względem lokalizacji arkusza CSS, a nie względem adresu dokumentu HTML!*

Polecenie pozwala wstawić dowolny obrazek jako tło pod elementem. Pamiętaj jednak, żeby obrazek w tle nie był zbyt jaskrawy. Spowoduje to tylko zaciemnienie zawartości strony i utrudni czytanie.

Wspólnie z tą własnością często podaje się dodatkowo kolor tła. W takim przypadku, jeśli obrazek będzie niedostępny lub użytkownik wyłączy wyświetlanie obrazów, tło przyjmie podany kolor. Ponadto jeżeli obrazek posiada przezroczystość (np. format \*.gif), zostanie ona zastąpiona podanym kolorem.

### Wielokrotne tło obrazkowe#

```
selektor { background-image: url(ścieżka dostępu do obrazka 1), url(ścieżka dostępu do obrazka 2), ... }
```

Dzięki powyższej składni możliwe jest wstawienie pod jednym elementem wielu obrazków w tle. Będą one ułożone jeden pod drugim - na wierzchu wyświetli się pierwsza z wymienionych grafik.

MSIE 8 w takim przypadku nie wyświetli żadnego obrazka w tle - nawet pierwszego z kolei!

## MARGINESY

### Górny margines {margin-top}

```
selektor { margin-top: rozmiar }
```

### Dolny margines {margin-bottom}

```
selektor { margin-bottom: rozmiar }
```

### Lewy margines {margin-left}

```
selektor { margin-left: rozmiar }
```

### Prawy margines {margin-right}



```
selektor { margin-right: rozmiar }
```

Selektorem może być praktycznie dowolny znacznik, np. p - akapit czy h1 - tytuł  
Natomiast jako "rozmiar" należy podać konkretną wartość w jednostkach długości. Wpisanie auto ustali wartość automatyczną.

Polecenie wprowadza dodatkowy odstęp między danym elementem a elementem poprzedzającym.

## Styl obramowania {border-...-style}

### Styl górnego obramowania:

```
selektor { border-top-style: styl }
```

### Styl dolnego obramowania:

```
selektor { border-bottom-style: styl }
```

### Styl lewego obramowania:

```
selektor { border-left-style: styl }
```

### Styl prawego obramowania:

```
selektor { border-right-style: styl }
```

We wszystkich przypadkach selektorem może być dowolny znacznik, np. p - akapit czy h1 - tytuł

Przykładowo, aby zmienić kolor tła oraz tekstu na całej stronie, możemy się posłużyć selektorem body, ponieważ właśnie wewnątrz znacznika <body>...</body> znajduje się cała zawartość strony. Zatem w arkuszu stylów wystarczy umieścić następujący kod:

```
body {
```

```
background-color: black;

color: white;

}
```

Zwróć uwagę, że deklaracja stylów, którą ujmuje się w nawiasy klamrowe, składa się tutaj z dwóch linijek. Każda z nich rozpoczyna się od podania tzw. cechy (inaczej własności), po której następuje wartość. Cecha określa, co chcemy zmienić w wyglądzie wybranego elementu, natomiast wartość - w jaki sposób ma się to zmienić.

Dla całej zawartości znacznika body (selektor) zmień kolor tła (cecha background-color) na czarny (wartość black) i kolor tekstu (cecha color) na biały (wartość white).

Pamiętaj, aby po wpisaniu cechy (własności) zawsze postawić znak dwukropka, a po każdej wartości - średnik. Zwróć również uwagę, że jeśli cecha (bądź wartość) zawiera znak myślnika (np. background-color), to przed nim ani po nim nie może znajdować się spacja. Aby ustalić inny kolor tła albo tekstu, wystarczy że w miejsce wartości wstawisz wybraną definicję koloru

### Kolory podstawowe

Oprócz kolorów zapisanych w systemie szesnastkowym, dostępna jest ograniczona paleta kolorów nazwanych, dla których zamiennie możemy używać ich nazwy zamiast kodu HEX:

Nazwa	HEX	Kolor
black	#000000	czarny
silver	#C0C0C0	srebrny
gray	#808080	szary
white	#FFFFFF	biały
maroon	#800000	kasztanowy
red	#FF0000	czzerwony
purple	#800080	purporowy
fuchsia	#FF00FF	fukcja
green	#008000	zielony
lime	#00FF00	limonowy
olive	#808000	oliwkowy
yellow	#FFFF00	żółty
navy	#000080	granatowy
blue	#0000FF	niebieski
teal	#008080	zielonomodry
aqua	#00FFFF	akwamaryna



## WYKAZY

### Typ stylu wykazu {list-style-type}

```
selektor { list-style-type: typ }
```

Selektorem mogą być znaczniki dotyczące wykazów: ul - wypunktowanie, ol - wykaz numerowany oraz li - pojedynczy punkt wykazu

Natomiast "typ" odpowiada za wygląd wyróżnika wykazu (markera) i należy zamiast niego wpisać:

**disc** - koło

**circle** - okrąg

**square** - kwadrat

**decimal** - liczby arabskie

**lower-alpha** - małe litery

**upper-alpha** - duże litery

### PRZYKŁAD

---

#### **list-style-type: disc**

- Punkt pierwszy
- Punkt drugi
- Punkt trzeci

## Ćwiczenia z działu HTML i CSS

1. Odszukaj w Internecie tabelę prezentującą aktualne kursy walut. Utwórz plik tabe-la.html i sporządź w nim podobną tabelę dla pięciu wybranych walut.
2. Utwórz stronę internetową, która odwzoruje układ treści z pliku wskazanego przez nauczyciela (np. przyroda.jpg).
3. Utwórz plik lista.html. Wypisz w nim uporządkowaną listę linków do pięciu stron z grami online oraz listę nieuporządkowaną z adresami trzech kanałów wideo.
4. Wykonaj stronę internetową, na której opiszesz swoje osiągnięcia. Każde z nich zapisz za pomocą nagłówka `<h3>` i krótko opisz, korzystając ze znacznika akapitu `<p>`.
5. Utwórz stronę WWW składającą się z trzech plików: index.html, omnie.html oraz kontakt.html. W każdym z tych plików napisz kilka zdań i wstaw listę nieuporządkowaną, w której aktywne będą linki tylko do pozostałych podstron.
6. Utwórz zewnętrzny szablon stylów i zapisz w nim właściwości dla elementu BODY: ciemnozielony kolor tła, biały kolor tekstu. Dla akapitu i nagłówka `<h4>` ustaw wyrównanie tekstu do środka, a dla elementu tworzącego tabelę - ramki w kolorze białym.
7. Zapisz kod tabeli o rozmiarach 5 x 5 bez widocznych krawędzi. Przygotuj obrazki o tej samej wielkości i umieść je w katalogu Images. W każdej komórce wstaw jeden obraz. Skorzystaj z atrybutów width i height.
8. Stwórz stronę WWW z kilkoma przepisami na swoje ulubione potrawy. Każdy przepis powinien zawierać opis, zdjęcie, listę produktów oraz kolejne kroki do wykonania.
9. Za pomocą zewnętrznego arkusza stylów sformatuj plik biuro\_podrozy.html tak, aby wyglądał w przeglądarce podobnie do rysunku wskazanego przez nauczyciela (np. biuro.jpg).

## 6 Projektowanie stron z wykorzystaniem WordPress

### Czym jest WordPress?

WordPress to **darmowy system zarządzania treścią typu open source** – CMS (z angielskiego Content Management System). Jest oparty na PHP i MySQL. WordPress jest rozwijany na licencji GNU GPL (tzw. Licencja wolnego oprogramowania). Ma szeroką społeczność użytkowników i programistów.

WordPress jest używany przez 30% z 10 milionów najpopularniejszych stron internetowych [statystyki z kwietnia 2018]. W sumie prowadzi **ponad 60 milionów stron internetowych**. WordPress został **założony** 27 maja 2003 r. przez Matta Mullenwega i Mika Littla.

WordPress **jest najpopularniejszym systemem wydawniczym dla stron internetowych i blogów**.

### Zalety WordPress

- **To nic nie kosztuje** – sam WordPress i wiele motywów i wtyczek są bezpłatne.
- **Nie wymaga umiejętności programistycznych** – użytkownik może zarządzać swoją stroną bez znajomości języków programowania, takich jak HTML, CSS i PHP.
- **Bezpłatne i częste aktualizacje** – społeczność WordPress często publikuje aktualizacje zabezpieczeń oraz nowe funkcje.
- **Wielojęzyczność** – rdzeń samego WordPressa oraz wiele wtyczek i motywów tłumaczonych jest na większość języków, w tym polski. Oto poradnik jak stworzyć wielojęzyczną stronę WordPress.
- Szereg **darmowych motywów** i modułów – korzyści WordPress z pewnością obejmują wiele istniejących dodatków.
- **Łatwość obsługi** – prosta administracja tworzeniem artykułów, praca z mediami i tworzenie treści.

- **Szeroka społeczność użytkowników** – istnieje wiele forów, na których eksperci społeczności chętnie Ci doradzą. Możesz także wziąć udział w oficjalnych spotkaniach o nazwie WordCamp i WordPress meetup.
- **Zmienność** – za pomocą motywów i modułów WP może być używany do wielu celów – osobisty blog, prezentacja firmy, handel elektroniczny, forum, a nawet sieć społecznościowa.
- Wsparcie dla większości hostingu – **nadal dobrym pomysłem jest sprawdzenie czy hosting spełnia optymalne wymagania do uruchomienia WP. Możesz zobaczyć mój samouczek dotyczący wyboru najlepszego hostingu WordPress.**
- Multisite – **możliwość tworzenia wielu witryn w jednej domenie.**

### Oficjalne strony społeczności WordPress

Aby lepiej zrozumieć, czym jest WordPress, możesz również przeglądać następujące strony internetowe:

- Oficjalna strona WordPress <http://www.wordpress.org/>
- Oficjalna polska strona <http://pl.wordpress.org/>
- Fundacja WordPress <http://wordpressfoundation.org/>

## 6.1 Instalacja WordPress

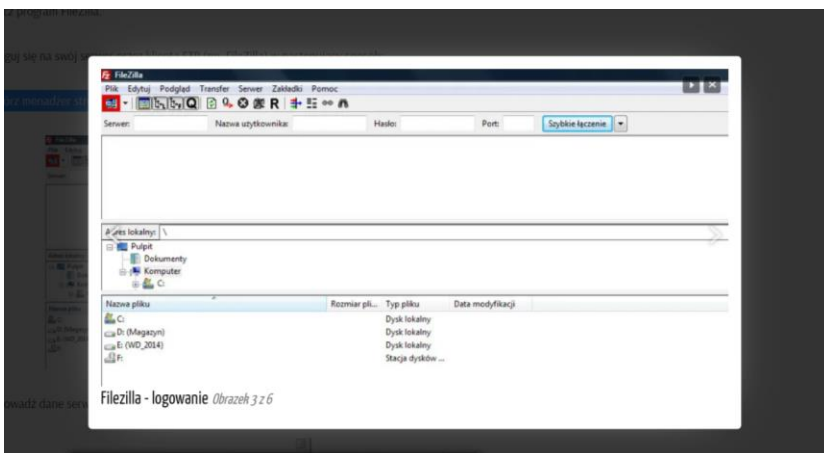
Aby zainstalować WordPressa:

1. Wejdź na stronę <http://pl.wordpress.org/>
2. Naciśnij przycisk Pobierz WordPress i zapisz plik na dysku.



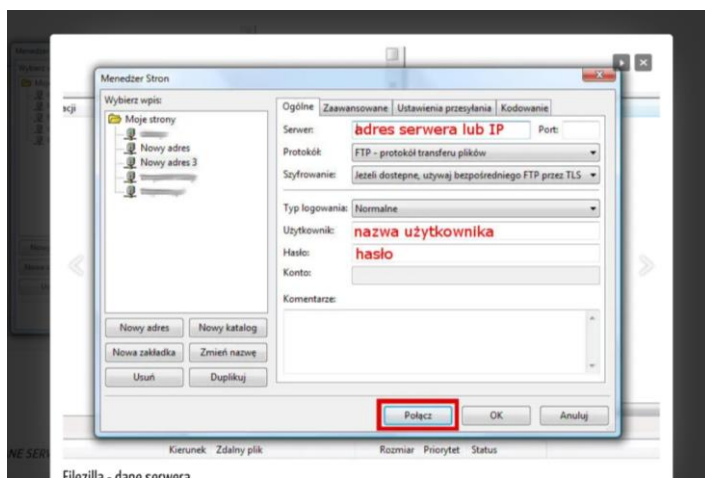


3. Wejdź do katalogu, w którym zapisałeś plik.
4. Rozpakuj ściągnięty plik.
5. Jeżeli posiadasz klienta FTP otwórz go, jeżeli nie – postępuj zgodnie z poniższymi wskazówkami:
6. Wejdź na stronę <http://www.filezilla.pl>.
7. Naciśnij niebieski przycisk z napisem „Download”.
8. Zapisz plik na swoim komputerze.
9. Zainstaluj program.
10. Włącz program FileZilla.
11. Zaloguj się na swój serwer przez klienta FTP (np. FileZilla) w następujący sposób:
  - a. Otwórz menadżer stron.



- b. Wprowadź dane serwera i naciśnij przycisk „Połącz”.

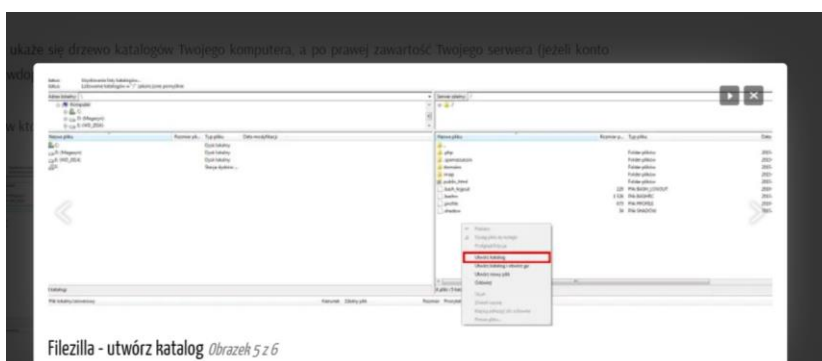




Dane dostępu do serwera (adres serwera, nazwa użytkownika i hasło) powinny przyjść w mailu informującym o rejestracji konta hostingowego. Jeżeli nie możesz ich odnaleźć, skontaktuj się z administratorem serwera. Hasło możesz także zmienić logując się do panelu administracyjnego swojego konta hostingowego.

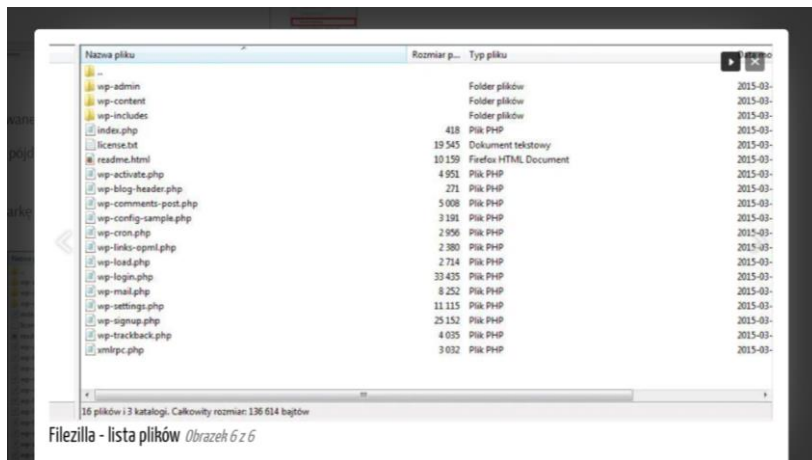
- c. Po lewej stronie ukaże się drzewo katalogów Twojego komputera, a po prawej zawartość Twojego serwera (jeżeli konto jest nowe najprawdopodobniej będzie pusto)

12. Utwórz katalog, w którym ma się znajdować strona (np. strona).



13. Skopiuj rozpakowane pliki na serwer (do utworzonego katalogu – np. strona).

14. Jeżeli wszystko pójdzie zgodnie z planem, w utworzonym katalogu (np. strona) powinny znajdować się następujące katalogi i pliki:



15. Otwórz przeglądarkę internetową (np. Mozilla Firefox lub Chrome) i wpisz następujący adres:

<http://adres-twojego-serwera/nazwa-utworzonego-katalogu>

16. Pojawi się ekran powitalny WordPress'a informujący o tym, że właśnie zaczynasz instalację i jakie dane musisz przygotować:

#### Potrzebne dane bazy danych:

- Nazwa bazy danych
- Nazwa użytkownika bazy danych
- Hasło użytkownika bazy danych
- Adres serwera bazy danych

17. Naciśnij przycisk „Zaczynamy” i wypełnij pierwsze trzy pola formularza (nazwa bazy danych, nazwa użytkownika, hasło) danymi bazy danych, które znajdziesz w panelu administracyjnym serwera. Jeżeli nie możesz ich znaleźć, skontaktuj się z administratorem swojego serwera.

W polu „Adres serwera bazy danych” pozostaw słowo „localhost” (odpowiednie dla zdecydowanej większości przypadków, choć niektóre firmy hostingowe wymagają wpisania innego adresu – w takim przypadku będziesz musiał skontaktować się z pomocą techniczną firmy hostingowej).

W polu „Prefiks tabel” pozostaw „wp\_” – innych prefiksów używa się w przypadku instalacji w jednej bazie danych wielu stron. My na razie poprzestaniemy na jednej.

18. Naciśnij przycisk „Wyślij”, a w kolejnym kroku „Uruchomienie instalacji”

19. Wypełni formularz dotyczący Twojej nowej strony internetowej:

20. Naciśnij przycisk „Zainstaluj WordPress”

21. To już koniec. W następnym kroku będziesz mógł zalogować się do panelu administracyjnego swojej nowej strony internetowej.

## 6.2 Obsługa panelu administracyjnego.

### Logowanie do panelu administratora

Strona logowania do panelu administratora znajduje się pod adresem:

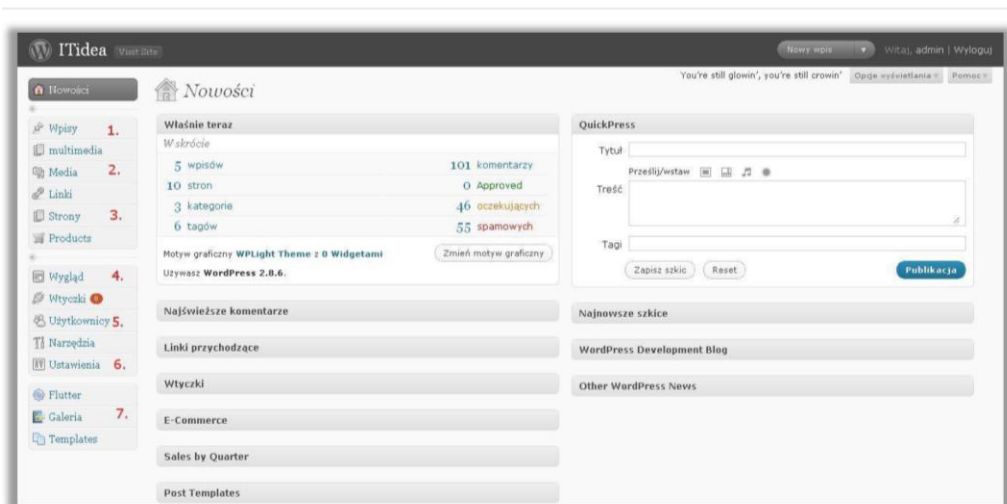
<http://Adres-Twojej-strony.pl/wp-admin>.

W oknie logowania dostępna jest funkcja przypomnienia hasła. Po jej wybraniu nowe hasło zostanie wysłane na Twój adres e-mail.

### Podstawowe funkcje panelu administracyjnego.

Po poprawnym zalogowaniu pojawia się następująca strona: Jej podstawowe elementy to:

1. Menu tworzenia i edycji wpisów i kategorii
2. Media
3. Menu tworzenia i edycji stron
4. Ustawienia wyglądu strony
5. Zarządzanie użytkownikami
6. Ustawienia ważnych elementów systemu zarządzania treścią
7. Galeria zdjęć



## Tworzenie i edycja wpisów i kategorii

Wpisy są artykułami, które mogą być wyświetlone na stronie w postaci listy tytułów i kilkunastu słów wstępu. Po naciśnięciu na tytuł, obrazek, lub przycisk „Więcej” następuje przejście do podstrony z pełną treścią. Wpisy można grupować stosując kategorie. W ten sposób można łatwo decydować jakie wpisy wyświetlać w danej liście.

### Tworzenie nowego wpisu

Aby utworzyć nowy wpis należy z podmenu „**Wpisy**” wybrać pozycję „**Dodaj nowy**”. W pole „**Wprowadź tytuł w tym miejscu**” należy wpisać tytuł wpisu, a w oknie edytora jego treść. W przypadku, gdy tworzony wpis jest artykułem publikowanym w dziale „Aktualności”, to przypisujemy go do kategorii „**Aktualności**”. W tym celu należy w sekcji „**Kategorie**” wybrać kategorię (zaznaczając pole obok jej nazwy). Zatwierdzić zmiany klikając przycisk „Opublikuj”.



Ekran edycji treści wpisu ma następujące elementy:

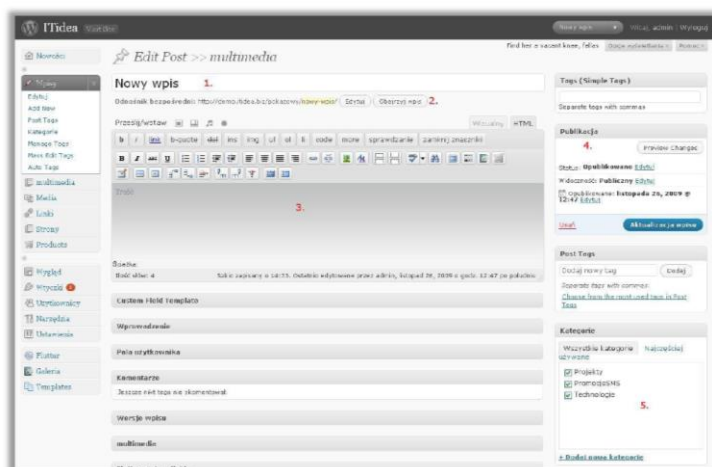
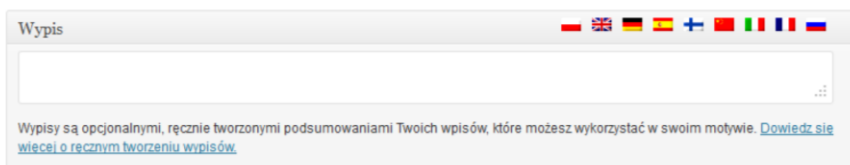
1. Tytuł.
2. Odnośnik bezpośredni – link do artykułu wyświetlany w pasku adresu przeglądarki – ważne, aby był opisowy, gdyż jego treść ułatwia wyszukiwanie strony przez wyszukiwarki.
3. Okno edycji treści wpisu – jego funkcjonalność przypomina itp. program MS Word.



4. Publikacja – tu można zdecydować czy dany wpis ma być widoczny, czyli opublikowany, czy też nie.
5. Kategorie – można wybrać ich wiele lub żadnej. Tutaj istnieje także możliwość utworzenia nowej kategorii dla wpisu.

## Dodawanie wpisu

Wypis pozwala wyświetlić podsumowanie lub początek tworzonego wpisu (przejsięcie do wpisu z pełną treścią następuje zwykle po kliknięciu odnośnika „więcej”).



## Wprowadzanie treści, wstawianie obrazków, galerii zdjęć

Panel edycji treści pozwala na swobodną i intuicyjną kontrolę nad treścią zawartą we wpisie. Jego funkcje są zbliżone do funkcji popularnych edytorów tekstu (jak Word). Dodatkowe funkcje są następujące:



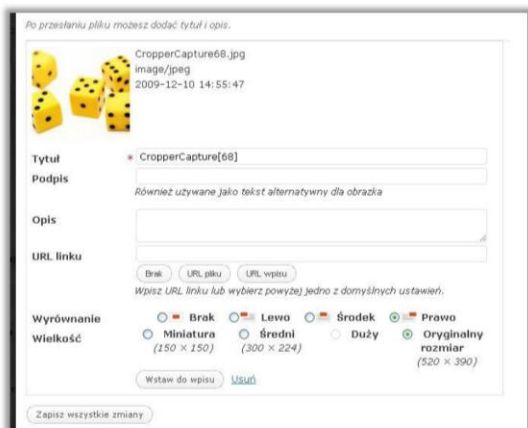
Przycisk ten służy wstawianiu obrazków



Po wciśnięciu wyświetli się następujące okno:



Za pomocą tego narzędzia możemy pobrać obrazek z własnego komputera (**przycisk „Z komputera”**), wstawić zewnętrzny adres obrazka (**„Z adresu URL”**), wybrać spośród pobranych już wcześniej obrazków (**„Biblioteka mediów”**) albo z galerii obrazków (**„NextGEN Gallery”**). Po wybraniu obrazka możemy zdecydować o jego ostatecznym wyglądzie:



Tytuł powinien być opisowy (ten napis pojawi się po najechaniu myszką na obrazek). W polu **„URL linku”** można wstawić odnośnik do innej strony.

**„Wyrównanie”** określa umieszczenie obrazka względem tekstu.

Ważnym elementem jest **„Wielkość”**. Obrazek na stronie nie powinien być za duży. W przypadku dużych zdjęć dobrym rozwiązaniem jest wybranie opcji **„Średni”** lub **„Duży”**. System stworzy wówczas kopię obrazka o odpowiednich rozmiarach (odwiedzający stronę mogą ściągnąć obrazek jedynie w tych rozmiarach – co zabezpiecza przed rozpowszechnianiem zdjęć dobrej jakości).

Przycisk „**Wstaw do wpisu**” kończy operację wstawiania obrazka.



Wstawianie galerii zdjęć.



Wstawianie linków. Funkcję tę należy stosować w przypadku umieszczania w treści odnośników do innych stron.



Ustalanie treści wstępu. Na stronie wyświetlające listę wpisów widoczny jest spis tytułów, towarzyszących im obrazków (jeżeli taka funkcja jest ustawiona), a także krótkiej treści pełniącej funkcję wstępu. Długość wstępu możemy ustalić za pomocą tego właśnie przycisku. Podzieli on tekst na dwie części.



Podział tekstu na strony

## Wstawianie zdjęcia –miniaturki dla wpisu lub strony

Często dla uatrakcyjnienia wyglądu strony tytułowi wpisowi lub stronie towarzyszy także obrazek – miniaturka. Wstawianiu takiego obrazka służy sekcja „**Ikona wpisu**”:



Po wciśnięciu przycisku „Ustaw ikonę wpisu” proces wyboru obrazka jest taki sam jak w przypadku wprowadzania zwykłych zdjęć (opisany wyżej). Aby wybrany obrazek stał się miniaturką należy na końcu wcisnąć „**Użyj jako ikony wpisu**”:





Wyrównanie  Brak  Do lewej  Do środka  Do prawej

Rozmiar  Miniatura (150 × 145)  
 Średni (300 × 114)  
 Duży  
 Pełny rozmiar (379 × 145)

[Użyj jako ikony wpisu](#) [Usuń](#)

## SEO – ułatwianie wyszukiwania treści przez wyszukiwarki internetowe

Na samym dole znajduje się okienko nazwane „**All In One SEO Pack**”. To ważny element służący ułatwieniu wyszukiwania strony przez wyszukiwarki internetowe. Jego elementy to tytuł, opis i słowa kluczowe. Odpowiedni dobór treści tu zawartej może znacząco wpłynąć na pozycję strony wśród rezultatów wyszukiwania w wyszukiwarkach. Treść tytułu i opisu jest wyświetlana w rezultatach wyszukiwania, dlatego ważne jest, aby w kilku słowach stanowiła trafny i zachęcający opis. Dobór słów kluczowych ma zasadniczy wpływ na późniejsze pozycjonowanie strony.

All in One SEO Pack  
[Click here for Support](#)

Title:

Description: 0 characters. Most search engines use a maximum of 160 chars for the description.

Keywords (comma separated):

Disable on this page/post:

## Zarządzanie kategoriami wpisów

Kategorie służą grupowaniu wpisów. Można je tworzyć podczas edytowania wpisu, albo wybierając „**Wpisy**” i „**Kategorie**”. Kategorie mogą być zagnieżdżone, tzn. istnieje możliwość tworzenia podkategorii przynależących do danej kategorii głównej. Aby dany wpis



należał do wybranej kategorii wystarczy w czasie edycji jego treści zaznaczyć odpowiednią nazwę w oknie kategorii:



Jak widać można zaznaczyć jedną, wiele lub żadnej kategorii. Jeżeli szablon Strony na to pozwala nazwy poszczególnych kategorii pojawią się w osobnym menu (obok menu ze stronami), a podkategorie będą widoczne jako rozwijane elementy tegoż menu.

## Media

Media to zbiór pobranych wcześniej przez nas obiektów (najczęściej zdjęć). To wygodny sposób, dzięki któremu nie trzeba wielokrotnie wczytywać tych samych plików. Funkcja „**Media**” umożliwi przegląd dostępnych zdjęć, dodawanie nowych oraz edycję i usuwanie istniejących.

## Tworzenie i edycja stron

Strony z technicznego punktu widzenia nie różnią się prawie niczym od wpisów. Ich zastosowanie jest jednak inne. Liczba stron jest zazwyczaj stała lub zmienia się nieznacznie. Treść w nich zawarta jest niezależna od ilości wpisów i kategorii. Dobrym pomysłem jest umieszczanie na stronach opisu „**O nas**”, informacji kontaktowych, informacji związanych z ofertą, czy cennikiem itp. W szablonie strony WWW istnieje odrębne menu zawierające



odnośniki do tych treści. Jeśli chodzi o wprowadzanie treści różnic względem wpisów jest mało. Jedną z niewielu jest następujące okno:



Umożliwia ono ustawienie własnej kolejności stron wyświetlanych w menu, a także ich hierarchię. Strony (podobnie jak kategorie) mogą być bowiem zagnieżdżone, czyli możliwe jest tworzenie podstron (podkategorii), co na stronie WWW może być widoczne jako menu rozwijane.

## Tworzenie nowej strony

Aby utworzyć nową stronę należy z podmenu „**Strony**” wybrać pozycję „**Dodaj nową**”. W pole „**Wprowadź tytuł w tym miejscu**” należy wpisać tytuł strony, a w oknie edytora jej treść. W przypadku, gdy tworzona strona jest stroną podrzędną w stosunku do innej, istniejącej już strony, to określamy jej rodzica. W tym celu należy w sekcji „**Atrybuty strony**” dla pozycji „**Rodzic**” wybrać właściwą stronę nadrzędną. Zatwierdzić zmiany klikając przycisk „**Opublikuj**”.



## Wygląd

W codziennej praktyce zazwyczaj nie zachodzi konieczność zmiany ustawień wyglądu strony WWW. Operacje dokonywane w tym panelu mogą zmienić część funkcjonalności witryny i zawsze trzeba się liczyć z ewentualną koniecznością „ręcznego dokonfigurowania” nowego szablonu. Narzędzie jest jednak dosyć wygodne w użyciu, a wprowadzone zmiany zazwyczaj da się cofnąć bez negatywnych konsekwencji.

## Użytkownicy

W tym miejscu można dodać nowe konto (np. dla współpracownika), które umożliwi innej osobie zalogowanie się do panelu administracyjnego korzystając z własnego loginu i hasła. Panel „**Użytkownicy**” pozwala także na zmianę własnych danych, jak imię i nazwiska, adres e-mail itp.

## Ustawienia

Narzędzie to pozwala na modyfikację podstawowych funkcji całego systemu strony WWW i panelu zarządzania treścią. Większość funkcji najlepiej pozostawić administratorowi, jednakże jest tu kilka elementów, na które warto mieć wpływ. Zazwyczaj jednak funkcje ustawień są zablokowane dla konta edytora i dostępne tylko dla administratora.

## Ustawienia ogólne

W tym miejscu możemy zdecydować o tym jaki ma być tytuł całej strony, format daty, godziny itp. Zmiana parametrów ustawień ogólnych nie wpłynie negatywnie na funkcjonowanie całości.



## Galeria zdjęć

To dosyć istotny element systemu zarządzania treścią. Mechanizm galerii ma wiele przydatnych funkcji i jednocześnie jest prosty w obsłudze.

### Tworzenie nowej galerii

Aby stworzyć nową galerię należy wybrać „Dodaj galerię/Obrazy” z menu „Galeria”, a następnie „Dodaj nową galerię”:

Dodaj nową galerię

Nowa galeria:

Utworzenie nowej, pustej galerii w katalogu `wp-content/gallery/`  
( Dozwolone znaki w nazwach katalogów i plików to: `a-z, A-Z, 0-9, -, _` )

[Dodaj galerię](#)

Nazwa galerii powinna w jasny sposób opisywać jej zawartość. Jest to szczególnie istotne w przypadku zarządzania wieloma galeriami. Nazwa galerii nie będzie wyświetlana na stronie.

### Dodawanie zdjęć do galerii

Po stworzeniu jednej lub wielu galerii, możemy przystąpić do pobrania zdjęć.

**UWAGA!:** Rozmiar zdjęcia nie może przekraczać 2272 x 1704 pikseli i dodatkowo należy pamiętać, aby wczytywany plik nie przekraczał objętości 2MB (dotyczy to także archiwów ZIP).

Najlepiej jakby obraz był mniejszy niż 300 - 500 KB, przy rozdzielczości 800 x 600 lub 1024 x 768. Do wyboru mamy:

- Przesłanie zdjęć w skompresowanym pliku ZIP
- Wybranie folderu za zdjęciami znajdującego się na serwerze
- Standardowe pobranie zdjęć z własnego komputera



Wybieramy „Prześlij obrazy”:

Prześlij obrazy

Prześlij obraz  Przeglądaj...

Przesłane obrazy umieść w  Wybierz galerię ▾

Note : Based on your server memory limit you should not upload larger images then 1280 x 960 pixel

Wyłącz przesyłanie plików za pomocą flash  **Prześlij obrazy**

CropperCapture[135].jpg

Przy wybieraniu zdjęć możemy zaznaczyć więcej niż jedno (np. całą zawartość katalogu). Ważne, aby przed wciśnięciem przycisku „Prześlij obrazy” wybrać z listy odpowiednią galerię. Zdjęcia zostaną pobrane automatycznie, jedno po drugim, a następnie zostaną stworzone ich miniaturki. Czasami operacja ta może potrwać kilka, kilkanaście minut.

## Zarządzanie galerią

Po wciśnięciu „Zarządzaj Galerią” zostanie wyświetlona lista dostępnych galerii. Możemy je usunąć lub edytować – naciskając na nazwie.

Panel edycji elementów wybranej galerii jest następujący:

Galeria : Ciekawe wydarzenie 12.12.2009

Ustawienia galerii (wzrost domyślny)

Tytuł: Ciekawe wydarzenie 12.12.2009 Podłącz do strony: Brak podłączenia

Opis:  Określ podjętki: 13 - woz-transportowy-820-ob

Ścieżka: wp-content/gallery/ciekawe-wydarzenie-12-12-2009 Autor: admin

Utwórz stronę: Strona główna (Bez strony nad) Dodaj stronę

Spójnij nowych obrazów **Zapisz zmiany**

Czynności  OK Sortuj galerię **Zapisz zmiany**

ID	Miniatura	Nazwa pliku	Treść alternatywna i tytuł / Opis	Tagi (oddzielone przecinkami)	Wyklucz
1		dom-w-dominu-932-obrazki-pl.jpg grudzień 11, 2009	dom-w-dominu-932-obrazki-pl.jpg		<input type="checkbox"/>
4		modliszka-karate-877-obrazki-pl.jpg grudzień 11, 2009	modliszka-karate-877-obrazki-pl.jpg		<input type="checkbox"/>
6		odwrocony-dom-294-obrazki-pl.jpg grudzień 11, 2009	odwrocony-dom-294-obrazki-pl.jpg		<input type="checkbox"/>

Umożliwia on m.in. zmianę tytułu, dodanie nowych zdjęć, usunięcie lub zablokowanie istniejących, ustalenie ich kolejności, opisu i tytułu, a nawet ręczne wykadrowanie obrazu miniaturki.

„**Treść alternatywna i tytuł / Opis**” to pola umożliwiające wpisanie tytułu (wyświetlanego po najechaniu na miniaturkę kursorem myszki). Poniżej jest miejsce na opis, który będzie wyświetlany pod wybranym - powiększonym zdjęciem. Ciekawą funkcją jest kadrowanie miniaturki. Automatycznie utworzone nie zawsze w pełni oddają charakter zamieszczonego zdjęcia (najczęściej dzieje się tak w przypadku zdjęć pionowych i poziomych miniatur). Po najechaniu myszką na zdjęcie pojawi się małe Menu. Wybieramy „**Edytuj miniaturkę**”, a następnie ustalamy, która część obrazu ma być zawarta w jego małej wersji:



## Ustawienia podstawowych funkcji galerii

Wybierając przycisk „**Opcje**” w menu „**Galeria**”, przechodzimy do centrum sterowania galeriami na naszej stronie. Zapisane tutaj ustawienia odnoszą się do wszystkich zbiorów obrazów przez nas utworzonych. Ciekawe funkcje to:


- „**Zintegrowany pokaz slajdów**”, czyli bardziej nowoczesna wersja programu umożliwiającego przeglądanie zdjęć w galerii. Można tę funkcję włączyć lub wyłączyć.
- „**Efekty**”, czyli sposób wyświetlania zdjęć. Najczęściej stosowanym efektem jest „**Thickbox**”, jednak można poeksperymentować z innymi dostępnymi. Po każdoraz-

zowej zmianie (i zachowaniu przyciskiem „**Zapisz zmiany**”) należy stronę WWW odświeżyć, aby zaobserwować zmiany.

- „**Znak wodny**”, czyli tekst lub logo, które ma być automatycznie wstawione w obrazki galerii. To dobry sposób chronienia autorstwa zdjęć. Znakiem wodnym może być obrazek (najlepiej czarno - biały). Przed zachowaniem zmian najlepiej upewnić się, za pomocą okna podglądu, czy uzyskaliśmy zamierzony efekt. Po zatwierdzeniu zmian nie można ich cofnąć, gdyż znak wodny zostanie wkomponowany w obrazki na stałe.
- „**Pokaz slajdów**” – jeżeli jest aktywny alternatywny sposób wyświetlania, możemy zdecydować jaki powinien być jego charakter.

### Wstawianie galerii zdjęć do wpisu lub strony

Galerię można wstawić w dowolnym miejscu tekstu strony lub wpisu. Należy w tym celu

ustawić kursor we właściwym miejscu w tekście, wcisnąć przycisk: , a następnie wybrać odpowiednią galerię z listy. W treści zostanie wówczas wstawiony znacznik: „**[nggallery id=15]**”, zawierający numer wybranej galerii. Oczywiście na stronie WWW nie będzie on widoczny

## 6.3 Obsługa motywów i wtyczek.

**Motyw (Theme)** to zbiór plików, które razem tworzą wygląd, układ i funkcjonalności strony zbudowanej na WordPressie. Motyw odpowiada za całkowity wygląd strony od wyglądu i rozmieszczenia menu, kolorystyki, typografii po układ strony. Każdy motyw jest inny i oferuje wiele możliwości zmiany wyglądu i funkcjonalności strony WordPress. Potocznie motywy często zwane są również szablonami lub skórkami.

Motywy są przygotowywane przez profesjonalnych programistów i grafików. Dodatkowo motywy są podzielone na dwie kategorie: motywy bezpłatne i premium. Wiele darmowych motywów znajduje się w katalogu motywów WordPress, czyli w tzw. repozytorium. Natomiast motywy premium można kupić w sklepach internetowych, od indywidualnych firm i programistów WordPress. Każdy motyw może być inny, oferując wiele możliwości zmiany wyglądu strony internetowej WordPress.

Używając WordPress możesz również tworzyć i rozwijać własne niestandardowe motywy.

### **Motywy bezpłatne.**

Bezpłatny motyw jak Underscores stworzony przez programistów WordPress lub inne frameworki stały się popularną podstawą dla nowych motywów.

To jak wygląda Twoja strona kolory, layout, ogólne ustawienie określa motyw. Każda strona internetowa WordPress wymaga obecności co najmniej jednego motywu, a każdy motyw powinien być zaprojektowany przy użyciu standardów WordPress ze strukturalnym PHP, poprawnym HTML i CSS. Używając WordPressa możesz instalować i przełączać się między różnymi motywami. Motywy umożliwiają zmianę wyglądu i funkcjonalności witryny WordPress bez zmiany kodu źródłowego lub treści witryny.

Motywami zarządzasz w kokpicie w sekcji „**Wygląd > Motywy**”.

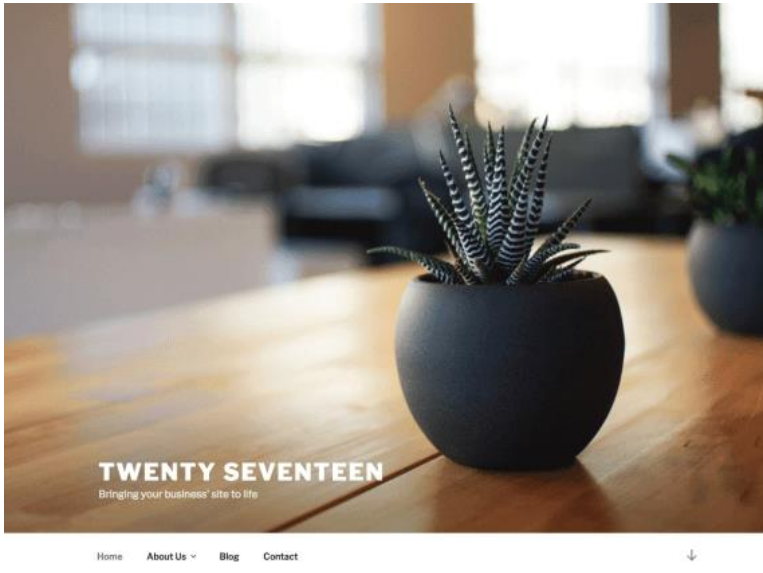
Zmianami wizualnymi strony dzięki funkcjom motywu możesz zarządzać w zakładce „**Dostosuj**” lub w „**Theme Options**” (opcje motywu).

Motywy można instalować bezpośrednio za pomocą panelu administracyjnego WordPress w zakładce „**Wygląd**” lub przez konto FTP w folderze „**wp-content/themes**”.



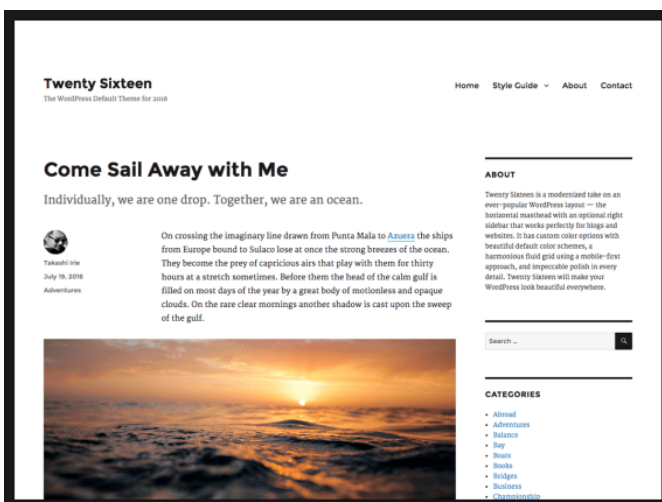


WordPress domyślnie pobierany jest i instalowany z trzema motywami:



### *Twenty Seventeen,*

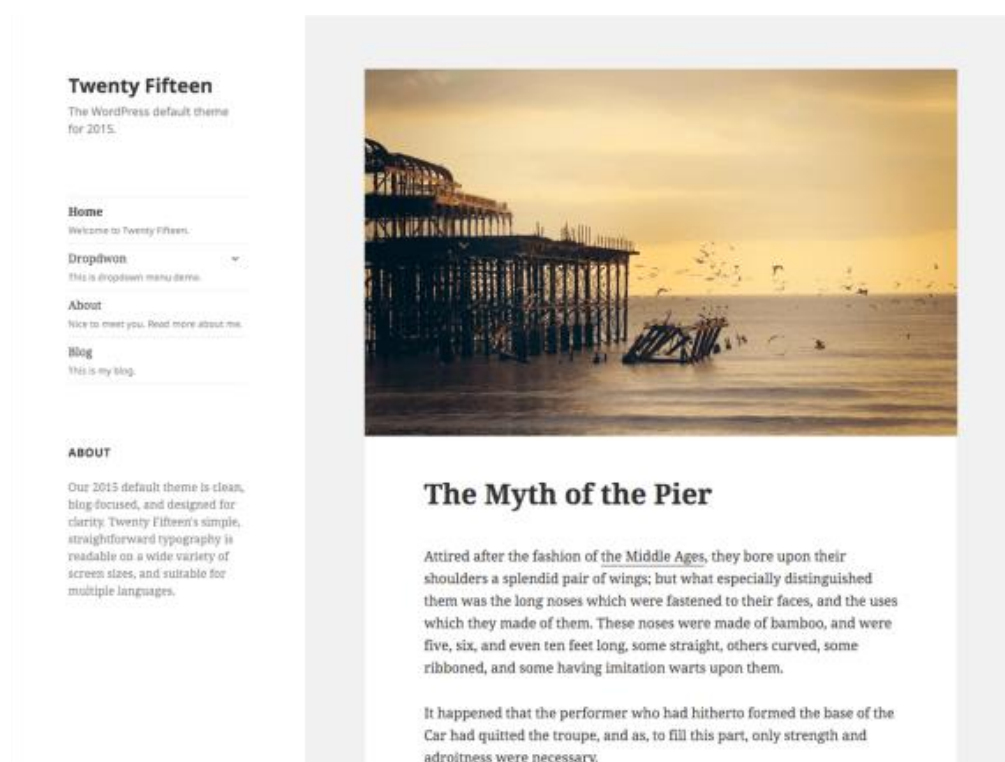
**Twenty Seventeen** dzięki filmom w nagłówku oraz obrazkom przynosi witrynie więcej życia. Motyw polecany jest do witryn biznesowych i oferuje wiele sekcji na stronie głównej, a także widżety, menu nawigacyjne i społecznościowe, logo i wiele innych. Spersonalizuj swój asymetryczną siatkę, użyj niestandardowej kolorystyki i prezentuj multimedialne treści z wykorzystaniem formatów. Nasz domyślny motyw dla 2017 działa świetnie w wielu językach, w każdych warunkach i na dowolnym urządzeniu.





## Twenty Sixteen,

**Twenty Sixteen** to unowocześniony, popularny wygląd WordPressa — poziomy nagłówek z opcjonalnym prawym panelem bocznym doskonale pasuje do blogów i innych witryn. Umożliwia ustawienie własnych kolorów, posiadając również gotowe propozycje zestawów kolorystycznych. Oparty jest na eleganckiej i elastycznej siatce podejścia mobile-first, będąc dopracowanym w każdym detalu. Twenty Sixteen sprawi, że WordPress będzie wyglądał pięknie — wszędzie

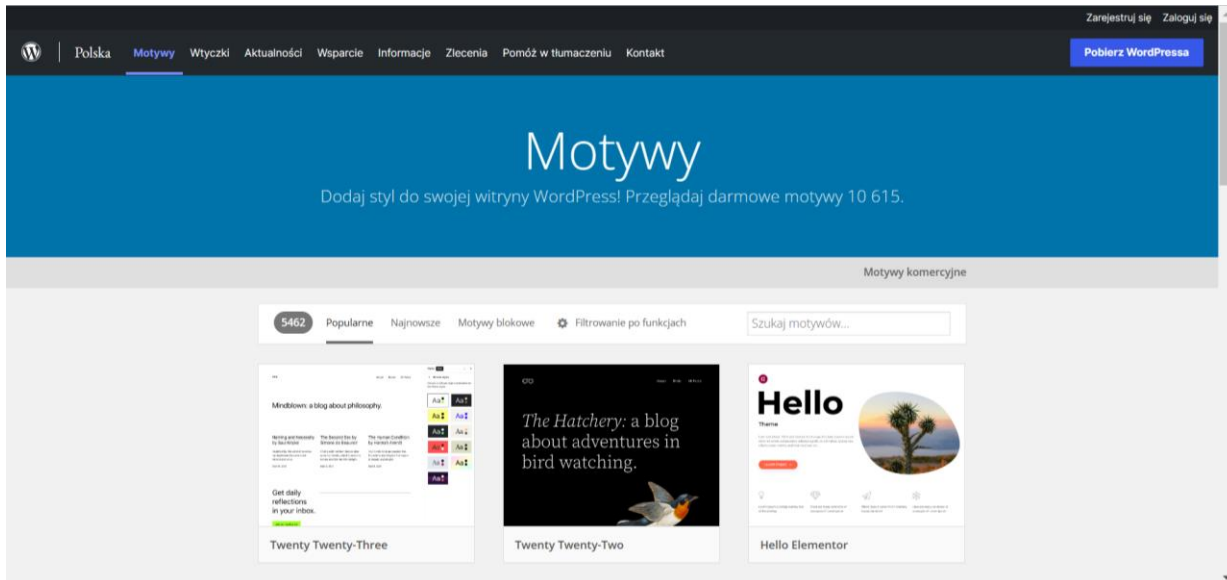


## Twenty Fifteen.

**Twenty Fifteen** - domyślny motyw (2015) jest schludny i przejrzysty, zorientowany na blogowanie. Motyw Twenty Fifteen posiada prostą i klarowną typografię, dzięki czemu jest czytelny na urządzeniach o różnych rozmiarach ekranów. Jest także przystosowany do wielu języków. Projektując motyw, kierowano się tzw. podejściem mobile-first, co oznacza, że najbardziej widoczna ma być zawsze treść, niezależnie od tego, czy odwiedzający używają do przeglądania smartfona, tabletu, laptopa, czy też komputera stacjonarnego.



Z poziomu kokpitu możesz przeglądać dostępne motywy w **repozytorium WordPress** lub wysłać na serwer kupiony wcześniej motyw premium.



Gdy znajdziesz odpowiedni motyw, kliknij przycisk **Zainstaluj**.

Jeżeli w zakładce **Motywy** masz zainstalowanych więcej skórek, możesz dowolnie zmieniać i aktywować nowy motyw na blogu.

Najedź kursorem na miniaturę motywu i kliknij **Aktywuj**.

Następnie przejdź do zakładki **Wygląd>Dostosuj**, w której skonfigurujesz bloga.

W panelu personalizacji **Dostosuj** możesz zmienić wygląd bloga zgodnie z możliwościami danego motywu.

## Płatne motywy WordPress

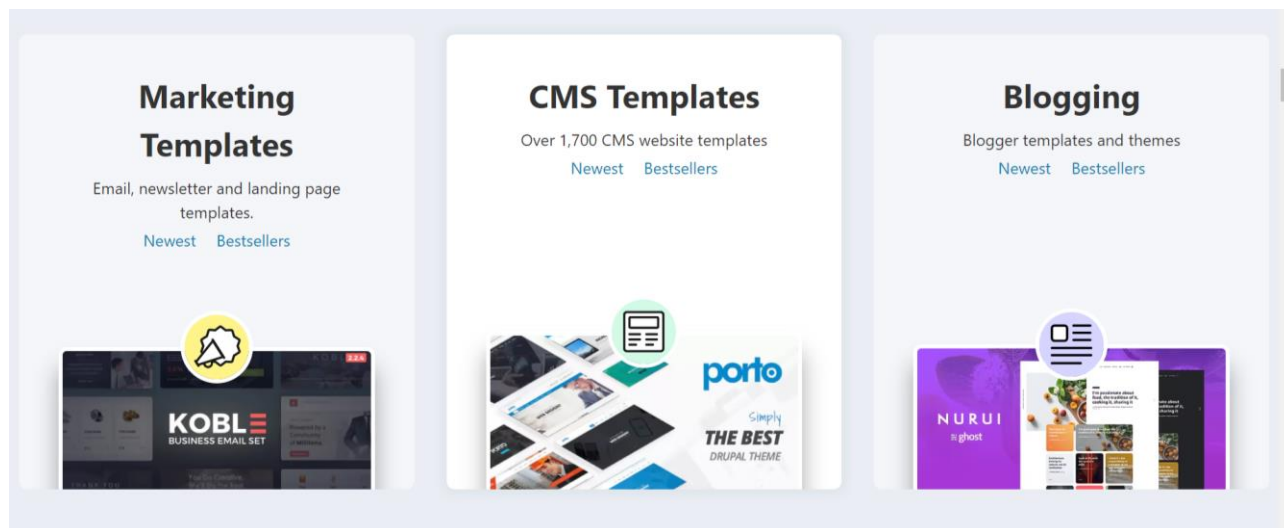
WordPress jest oparty na bezpłatnej licencji GDL i takie też muszą być motywy. Mimo tysięcy motywów dostępnych w katalogu WordPressa, niektórzy użytkownicy preferują motywy płatne ze wsparciem technicznym, za które są skłonni zapłacić. Istnieje wiele firm i sklepów internetowych komercyjnie wspierających motywy na licencji GPL, świadcząc do nich jednocześnie dodatkowe, płatne usługi.

Licencja GPL nie oznacza, że wszystkie produkty muszą być darmowe, natomiast oznacza, że otrzymujesz oprogramowanie, które nie może ograniczać Twojej wolności podczas jego używania.

- Płatne motywy polecane przez WordPress.org.
- Płatne motywy od sklepu Theme Forest.

możesz również szukać motywów w:

- Płatne motywy lub darmowe motywy od sklepu Template Monster.
- Płatne motywy od sklepu Creative Market.



## Wtyczki

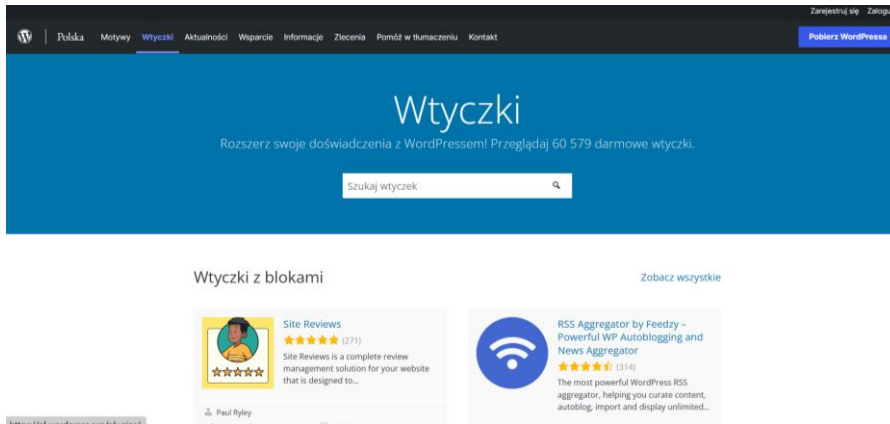
Wtyczki to dodatki instalowane w systemie WordPress. Możliwości oraz zastosowanie pluginów są ogromne, od prostego i szybkiego dodania przycisku „Lubię to” na stronie WWW, skończywszy na całym silniku np. sprzedaży produktów czy dodatków w postaci wbudowanych edytorów tekstu, systemów logowania, galerii zdjęć itp.

Wtyczki pozwalają także na optymalizację stron WWW. Dzięki nim możemy skorzystać z gotowych rozwiązań, które zadbają o poprawność wyświetlania kodu strony, optymalizację jego działania, a także pozycjonowanie. W tym zakresie WordPress nie ma sobie równych. Wachlarz zastosowań wtyczek WordPress daje nam nieograniczone możliwości budowania stron WWW, a samo ich uruchomienie sprowadza się zwykle do kliknięcia przycisków **Zainstaluj i Włącz**, dostępnych w panelu administratora systemu CMS.

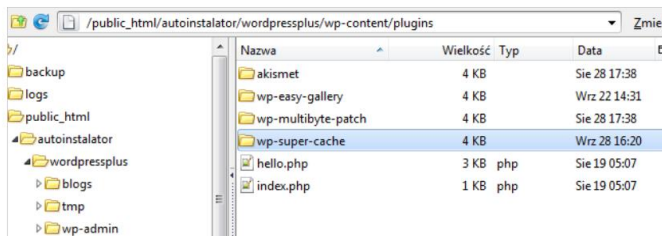


## Ręczna instalacja wtyczki na serwerze FTP

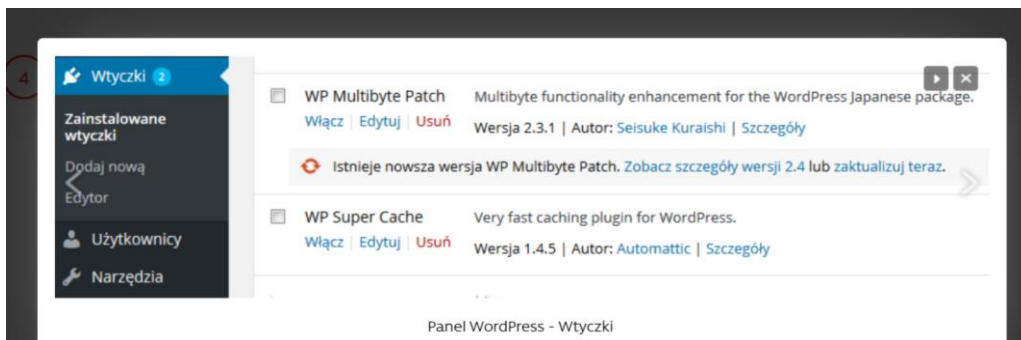
1. Pobierz wybraną **wtyczkę** ze strony autora lub bazy wtyczek. Rozpakuj archiwum na dysku komputera.



2. Połącz się z serwerem FTP przy pomocy **klienta FTP**.
3. Umieść pliki wtyczki w katalogu ze stroną WordPress, w katalogu **/wp-content/plugins**



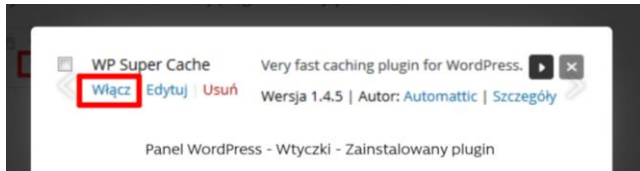
4. Zaloguj się do **panelu administratora WordPress**, a następnie przejdź do opcji menu **Wtyczki**.







5. Zainstalowany plugin będzie dostępny na liście zainstalowanych dodatków, skorzystaj z przycisku **Włącz**, aby uruchomić zainstalowany plugin na swojej stronie.



### Lista popularnych wtyczek WordPress

Lista najczęściej polecanych i wybieranych wtyczek WordPress, które pozwolą rozbudować stronę WWW, a także zadbać o jej zawartość, optymalne działanie, bezpieczeństwo i wygląd. Lista wtyczek WordPress została utworzona na podstawie własnych doświadczeń oraz opinii naszych klientów opublikowanych na forum.home.pl.

### Wtyczka WordPress — zawartość

- *Contact Form 7 (formularze)* - jedno z najczęściej stosowanych przez użytkowników narzędzi do tworzenia formularzy kontaktowych. Prosty, ale bardzo elastyczny w modyfikacji.
- *Intuitive Custom Post Order* - pozwala porządkować wpisy (Posts, Pages, and Custom Post Types and Custom Taxonomies) za pomocą funkcji „Drag and Drop”. Dzięki wtyczce można samodzielnie definiować kolejność artykułów/wpisów na ich liście.
- *Category Order and Taxonomy Terms Order* - pozwala porządkować kategorię wpisów według samodzielnie ustalonej kolejności (analogicznie jak wtyczka powyżej).
- *Redirection* - pozwala zarządzać przekierowaniami (np. typu 301) poprzez panel administracyjny WordPress oraz pozwala na monitorowanie występujących błędów 404.
- *Broken Link Checker* — wtyczka sprawdza na bieżąco opublikowane posty, komentarze i inne treści pod względem nieaktywnych linków lub obrazów. W łatwy sposób pozwala naprawić nieaktywne linki.

## Ćwiczenie na koniec działu .

### Projekt zespołowy

#### **Wykonanie i opublikowanie w Internecie responsywnej strony internetowej z wykorzystaniem systemu CMS.**

#### Cel projektu

Strony internetowe wykorzystuje się dziś w niemal każdej sferze działalności człowieka. Dzięki istniejącym narzędziom stworzenie takiej strony jest w zasięgu każdego. Od nowoczesnych stron użytkownicy wymagają, aby były czytelne na ekranach różnej wielkości. W tym projekcie utworzycie właśnie taką stronę internetową.

#### Propozycje tematów:

1. Przyroda wokół nas
2. Wynalazki, które zmienimy świat
3. Zdrowe odżywianie
4. Portal klasowy
5. Recenzje książek, albumów muzycznych, filmów i seriali.

**Wybierzcie temat projektu strony WWW** - najlepiej taki, który odpowiada waszym zainteresowaniom (możecie skorzystać z podanych wyżej propozycji). Temat dobierzcie tak, aby treść, którą zaprezentujecie, była bogata merytorycznie | wizualnie.

#### Przemyślcie | przygotujcie konspekt opisujący:

- strukturę (menu) witryny,
- kolorystykę strony | wygląd graficzny,
- tematykę strony,
- tematykę poszczególnych wpisów na stronie,
- podział ról w zespole tworzącym stronę — należy uwzględnić role: projektanta strony, administratora, autora wpisów, redaktora, grafika.

**Utwórzcie konto na stronie <https://wordpress.com>** i korzystając z przygotowanego wcześniej konspektu, wykonajcie następujące czynności:

- a. Odpowiedzcie na podstawowe pytania dotyczące tytułu strony, zawartości i stopnia opanowania systemu WordPress,
- b. wybierzcie nazwę witryny i domeny, pod która zostanie zarejestrowana – możecie skorzystać z darmowej opcji i wybrać domenę [wordpress.com](https://wordpress.com),
- c. ustalcie plan taryfowy — w zakładce Plany taryfowe możecie wybrać przycisk **Zacznij za darmo**,
- d. zgodnie z konspektem wytypujcie administratora serwisu, który poda adres e-mail i ustali hasło do panelu administracyjnego CMS.

**Utwórzcie konta pozostałych członków grupy** i przydzielcie im role zgodnie z konspektem.

**Wprowadźcie ustalenia graficzne** i dostosujcie układ strony do waszych założeń.

**Przetestujcie opcje wyświetlania serwisu** na różnych urządzeniach i wymieńcie się spostrzeżeniami.

Nanieście poprawki na podstawie uwag, które zbierzecie.

Wypełnijcie serwis treścią.

Kiedy strona będzie gotowa, udostępnicie jej adres nauczycielowi.

### **Przykładowe narzędzia**

**WordPress** — system CMS, który pełni również funkcje hostingu (domena [wordpress.com](https://wordpress.com)). Pozwala na tworzenie responsywnych stron internetowych.

**Edytor tekstu** - przydatny przy opracowywaniu konspektu witryny. Konspekt powinien mieć cechy dokumentu wielostronicowego — wprowadzony podział na nagłówki i spis treści.





### Źródła:

- *Bujnowski, Z. Talaga, Informatyka, t. 1, WSzPWN, Warszawa 2003.*
- *J. Mazur, P. Perekietka, Z. Talaga, J. S. Wierzbicki Informatyka na czasie 1 poziom rozszerzony Nowa Era 2020*
- *M. Borowiecki Informatyka na czasie 2 poziom rozszerzony 2 Nowa Era 2020*
- *E. Gurbiel, G. Hardt-Olejniczak, E. Kołczyk, H. Krupicka, M. Sysło, Informatyka, cz. 1, WSiP, Warszawa 2002.*
- *G. Koba, Informatyka, cz. 1, Migra, Wrocław 2003.*
- <https://www.math.edu.pl/system-pozycyjny>
- <https://stronawordpress.com/poradnik-wordpress/>