# Informatyka 1 (ES1E2009)

Politechnika Białostocka - Wydział Elektryczny

Elektrotechnika, semestr II, studia stacjonarne I stopnia

Rok akademicki 2019/2020

## Wykład nr 1 (25.02.2020)

dr inż. Jarosław Forenc

# Dane podstawowe

- dr inż. Jarosław Forenc
- Politechnika Białostocka, Wydział Elektryczny,
  Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki
  ul. Wiejska 45D, 15-351 Białystok
  WE-204
- e-mail: j.forenc@pb.edu.pl
- tel. (0-85) 746-93-97
- http://jforenc.prv.pl
  - Dydaktyka - dodatkowe materiały do pracowni specjalistycznej
- konsultacje:
  - poniedziałek, godz. 09:00-10:00, WE-204
  - wtorek, godz. 11:00-12:00, WE-204
  - środa, godz. 09:00-10:00 + 12:15-13:00, WE-204
  - piątek, godz. 17:00-18:30, WE-204 (studia zaoczne)

# Program wykładu (1/2)

1. Informacja analogowa i cyfrowa. Pozycyjne i niepozycyjne systemy liczbowe. Konwersje pomiędzy systemami liczbowymi.
2. Jednostki informacji cyfrowej. Kodowanie informacji. Kodowanie znaków.
3. Kodowanie liczb. Reprezentacja liczb w systemach komputerowych: stałoprzecinkowa i zmiennoprzecinkowa. Standard IEEE 754.
4. Programowanie w języku C. Deklaracje i typy zmiennych, operatory i wyrażenia arytmetyczne, operacje wejścia-wyjścia, operatory relacyjne i logiczne, wyrażenia logiczne, instrukcja warunkowa if, instrukcja switch, operator warunkowy, pętle (for, while, do .. while), tablice jednowymiarowe.

# Program wykładu (2/2)

5. Architektura komputerów. Klasyfikacja systemów komputerowych (taksonomia Flynna). Architektura von Neumana i architektura harwardzka.
6. Budowa i zasada działania komputera. Procesor, pamięć wewnętrzna i zewnętrzna. Komunikacja z urządzeniami zewnętrznymi, interfejsy komputerowe.
7. Algorytmy. Definicja algorytmu. Klasyfikacje i sposoby przedstawiania algorytmów. Rekurencja. Złożoność obliczeniowa. Sortowanie. Klasyfikacje algorytmów sortowania. Wybrane algorytmy sortowania.
8. Zaliczenie wykładu.

# Literatura (1/2)

1. R. Kawa, J. Lembas: „Wykłady z informatyki. Wstęp do informatyki”. PWN, Warszawa, 2017.
2. W. Kwiatkowski: „Wprowadzenie do kodowania”. BEL Studio, Warszawa, 2010.
3. S. Gryś: „Arytmetyka komputerów w praktyce”. PWN, Warszawa, 2013.
4. W. Stallings: „Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność”. WNT, Warszawa, 2004.
5. A.S. Tanenbaum: „Strukturalna organizacja systemów komputerowych”. Helion, Gliwice, 2006.
6. K. Wojtuszkiewicz: „Urządzenia techniki komputerowej. Część 1. Jak działa komputer? Część 2. Urządzenia peryferyjne i interfejsy”. PWN, Warszawa, 2011.

# Literatura (2/2)

7. K. Banasiak: „Algorytmizacja i programowanie w Matlabie”. Wydawnictwo BTC, Legionowo, 2017.
8. P. Wróblewski: „Algorytmy, struktury danych i techniki programowania. Wydanie V”. Helion, Gliwice, 2015.
9. M. Sysło: „Algorytmy”. Helion, Gliwice, 2016.
10. B. Buczek: „Algorytmy. Ćwiczenia”. Helion, Gliwice, 2008.
11. G. Coldwin: „Zrozumieć programowanie”. PWN, Warszawa, 2015.
12. S. Prata: „Język C. Szkoła programowania. Wydanie VI”. Helion, Gliwice, 2016.

# Efekty uczenia się i system ich oceniania

Podstawę do zaliczenia przedmiotu (uzyskanie punktów ECTS) stanowi stwierdzenie, że każdy z założonych efektów uczenia się został osiągnięty w co najmniej minimalnym akceptowalnym stopniu.

| | |
|---|---|
| **EU1** | identyfikuje i opisuje zasadę działania podstawowych elementów systemu komputerowego |

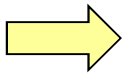
| | |
|---|---|
| **EU2** | formułuje algorytmy komputerowe rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice |

# Zaliczenie wykładu - efekty uczenia się (EU1)

- Student, który zaliczył przedmiot:

  identyfikuje i opisuje zasadę działania podstawowych elementów systemu komputerowego

- Student, który zalicza na ocenę dostateczny (3):
  - wymienia podstawowe elementy systemu komputerowego i podaje ich przeznaczenie
  - krótko charakteryzuje klasyfikację Flynna systemów komputerowych
  - wyjaśnia podstawowe pojęcia związane z architekturą i zasadą działania systemów komputerowych
  - dokonuje konwersji liczby całkowitej bez znaku z systemu dziesiętnego na system o dowolnej podstawie i z systemu o dowolnej podstawie na system dziesiętny



# Zaliczenie wykładu - efekty uczenia się (EU1)

- Student, który zalicza na ocenę dostateczny (3) (c.d.):
  - wyjaśnia na czym polega zapis zmiennoprzecinkowy liczby rzeczywistej oraz postać znormalizowana tego zapisu
- Student, który zalicza na ocenę dobry (4) (oprócz wymagań na ocenę 3):
  - opisuje strukturę i zasadę działania wybranych elementów systemu komputerowego
  - wymienia różnice pomiędzy architekturą von Neumana i architekturą harwardzką systemów komputerowych
  - dokonuje konwersji liczby całkowitej ze znakiem na wybrany kod (ZM, U1, U2) i odwrotnie
  - charakteryzuje wybrane kody liczbowe (NKB, BCD, Graya) i alfanumeryczne (ASCII, ISO-8859, Unicode)

# Zaliczenie wykładu - efekty uczenia się (EU1)

- Student, który zalicza na ocenę bardzo dobry (5) (oprócz wymagań na ocenę 4):
  - przedstawia cel stosowania oraz zasadę działania pamięci podręcznej
  - omawia sposób kodowania wartości specjalnych w standardzie IEEE 754

# Zaliczenie wykładu - efekty uczenia się (EU2)

- Student, który zaliczył przedmiot:

  formułuje algorytmy komputerowe rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice

- Student, który zalicza na ocenę dostateczny (3):
  - przedstawia rozwiązanie prostego problemu w postaci schematu blokowego opisującego algorytm komputerowy
  - podaje definicję algorytmu komputerowego i wymienia metody opisu algorytmów
  - przedstawia sposób sortowania wektora liczb stosując wybraną, prostą metodę sortowania

# Zaliczenie wykładu - efekty uczenia się (EU2)

- Student, który zalicza na ocenę dobry (4) (oprócz wymagań na ocenę 3):
  - przedstawia rozwiązanie złożonego problemu w postaci schematu blokowego opisującego algorytm komputerowy
  - wyjaśnia pojęcie złożoności obliczeniowej algorytmu, podaje złożoności obliczeniowe przykładowych algorytmów
- Student, który zalicza na ocenę bardzo dobry (5) (oprócz wymagań na ocenę 4):
  - wyjaśnia pojęcie rekurencji i podaje przykłady algorytmów rekurencyjnych
  - przedstawia sposób sortowania wektora liczb stosując metodę sortowania szybkiego (Quick-Sort)

# Zaliczenie wykładu

- Sprawdzian pisemny:
  - sprawdzian: 16.06.2020 (wtorek), godz. 12:15, WE-Aula III
  - poprawa: termin do ustalenia
- Na zaliczeniu oceniane będą dwa efekt uczenia się (EU1, EU2)
- Za każdy efekt uczenia się można otrzymać od 0 do 100 pkt.
- Na podstawie otrzymanych punktów wystawiana jest ocena:

| Punkty | Ocena | Punkty | Ocena |
|---|---|---|---|
| 91 - 100 | 5,0 | 61 - 70 | 3,5 |
| 81 - 90 | 4,5 | 51 - 60 | 3,0 |
| 71 - 80 | 4,0 | 0 - 50 | 2,0 |

# Zaliczenie wykładu

- Prowadzący zajęcia może przyznawać dodatkowe punkty za aktywność na wykładzie
- Ocena końcowa wyznaczana jest na podstawie sumy otrzymanych punktów:

| Punkty | Ocena | Punkty | Ocena |
|---|---|---|---|
| 182 - 200 | 5,0 | 122 - 141 | 3,5 |
| 162 - 181 | 4,5 | 102 - 121 | 3,0 |
| 142 - 161 | 4,0 | 0 - 101 | 2,0 |

# Terminy zajęć

- Wykład nr 1 - 25.02.2020
- Wykład nr 2 - 10.03.2020
- Wykład nr 3 - 24.03.2020
- Wykład nr 4 - 07.04.2020
- Wykład nr 5 - 05.05.2020
- Wykład nr 6 - 19.05.2020
- Wykład nr 7 - 02.06.2020
- Wykład nr 8 - 16.06.2020 (1 h, 12:15-13:00)

# Plan wykładu nr 1

- Język C
  - historia, struktura programu
  - kompilacja, zapis kodu
  - sekwencje sterujące, komentarze
- Pojęcia: informatyka i informacja
- Informacja analogowa i cyfrowa
- Systemy liczbowe
  - liczby i cyfry
  - systemy pozycyjne (dziesiętny, dwójkowy, szesnastkowy)
  - systemy niepozycyjne (rzymski)
- Konwersje między systemami liczbowymi

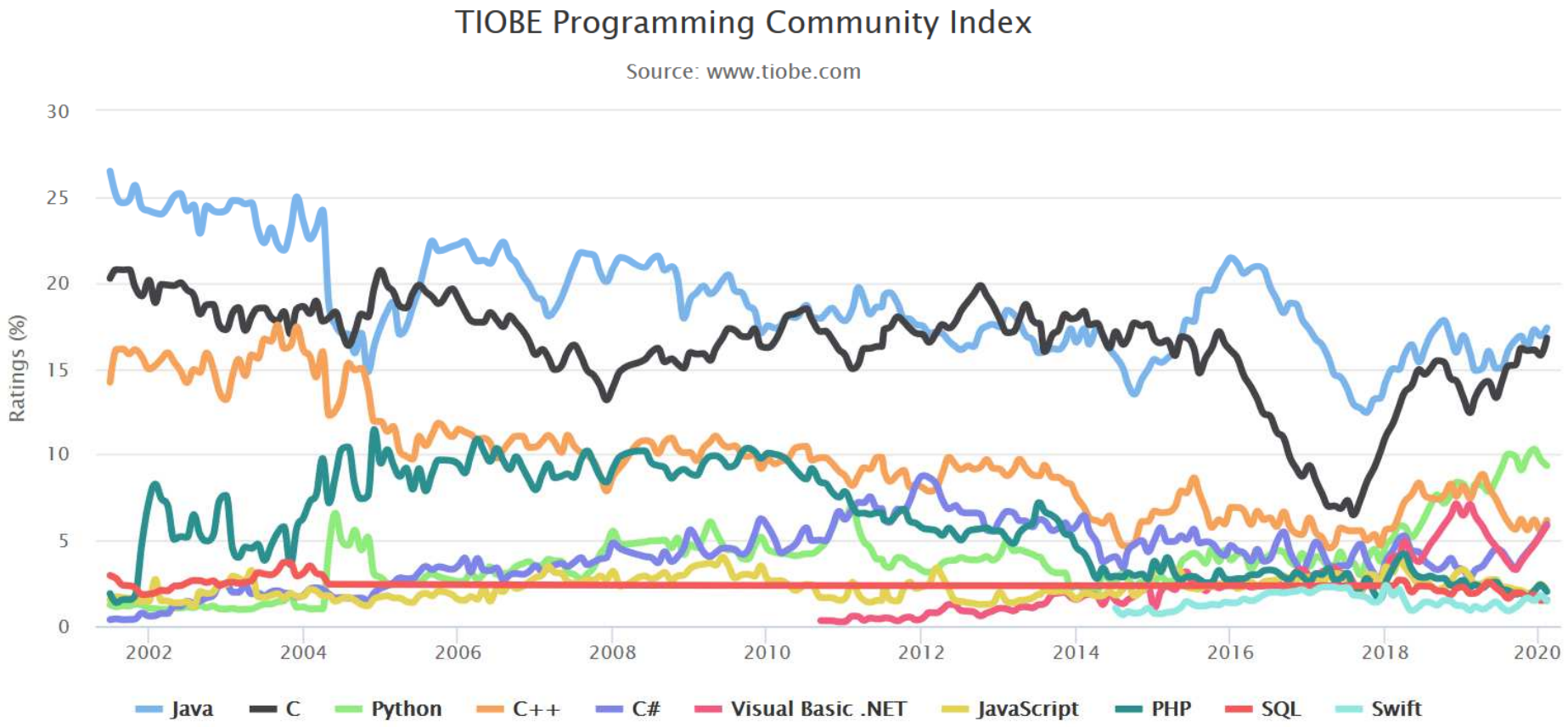
# Język C - krótka historia (1/2)

- 1969 - język BCPL - Martin Richards, University Mathematical Laboratories, Cambridge
- 1970 - język B - Ken Thompson, adaptacja języka BCPL dla pierwszej instalacji systemu Unix na komputer DEC PDP-7
- 1972 - język NB (New B), nazwany później C - Dennis Ritchie, Bell Laboratories, New Jersey, system Unix na komputerze DEC PDP-11
  - 90% kodu systemu Unix oraz większość programów działających pod jego kontrolą napisane w C
- 1978 - książka „The C Programming Language” (Kernighan, Ritchie), pierwszy podręcznik, nieformalna definicja standardu (K&R)

# Język C - krótka historia (2/2)

- 1989 - standard ANSI X3.159-1989 „Programming Language C” (ANSI C, C89)
- 1990 - adaptacja standardu ANSI C w postaci normy ISO/IEC 9899:1990 (C90)
- 1999 - norma ISO/IEC 9899:1999 (C99)
- 2011 - norma ISO/IEC 9899:2011 (C11)
- 2018 - norma ISO/IEC 9899:2018 (C18 lub C17)

# Język C - TIOBE Programming Community Index



# Język C - pierwszy program

- Niesformatowany plik tekstowy o odpowiedniej składni i mający rozszerzenie .c
- Kod najprostszego programu:

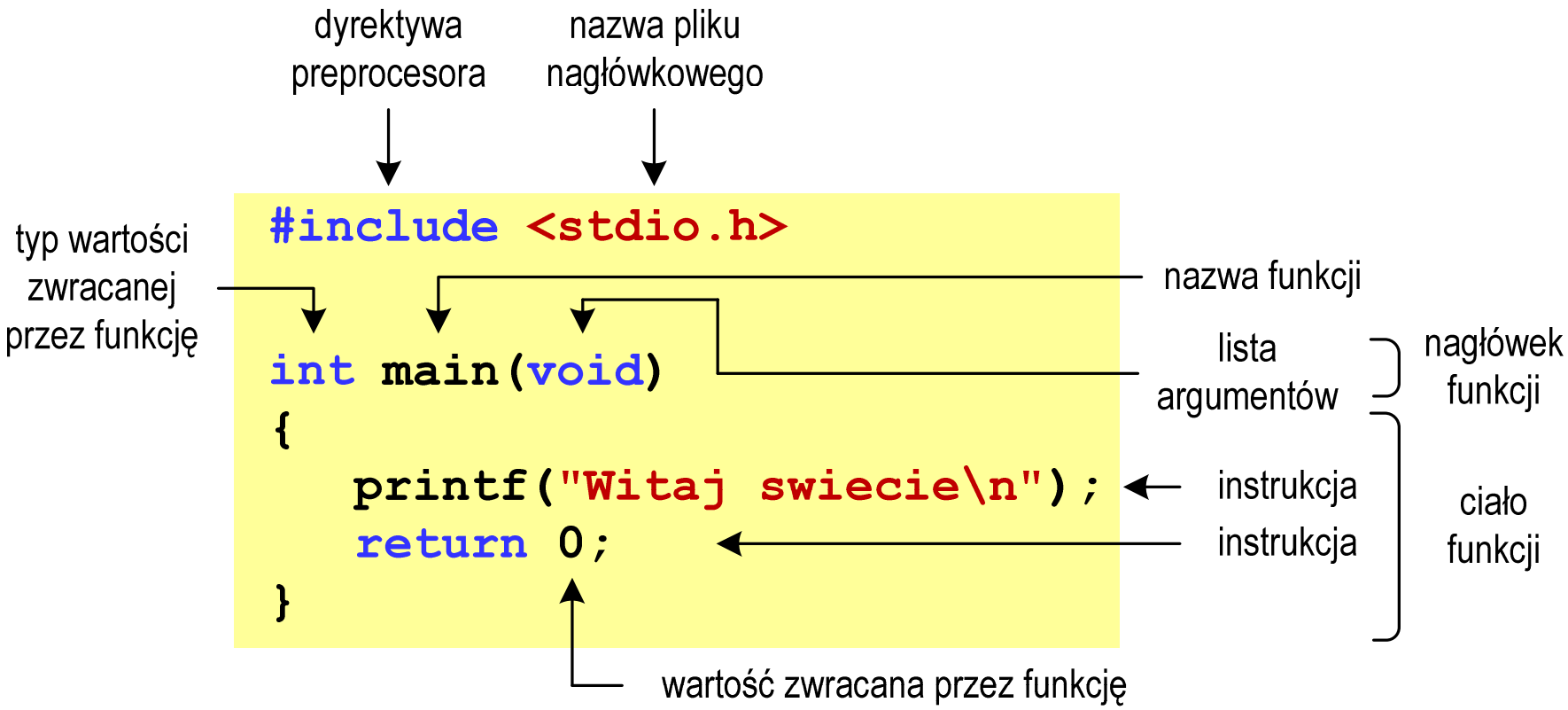
```
#include <stdio.h>

int main(void)
{
    printf("Witaj swiecie\n");
    return 0;
}
```

- Program konsolowy - wyświetla w konsoli tekst Witaj swiecie

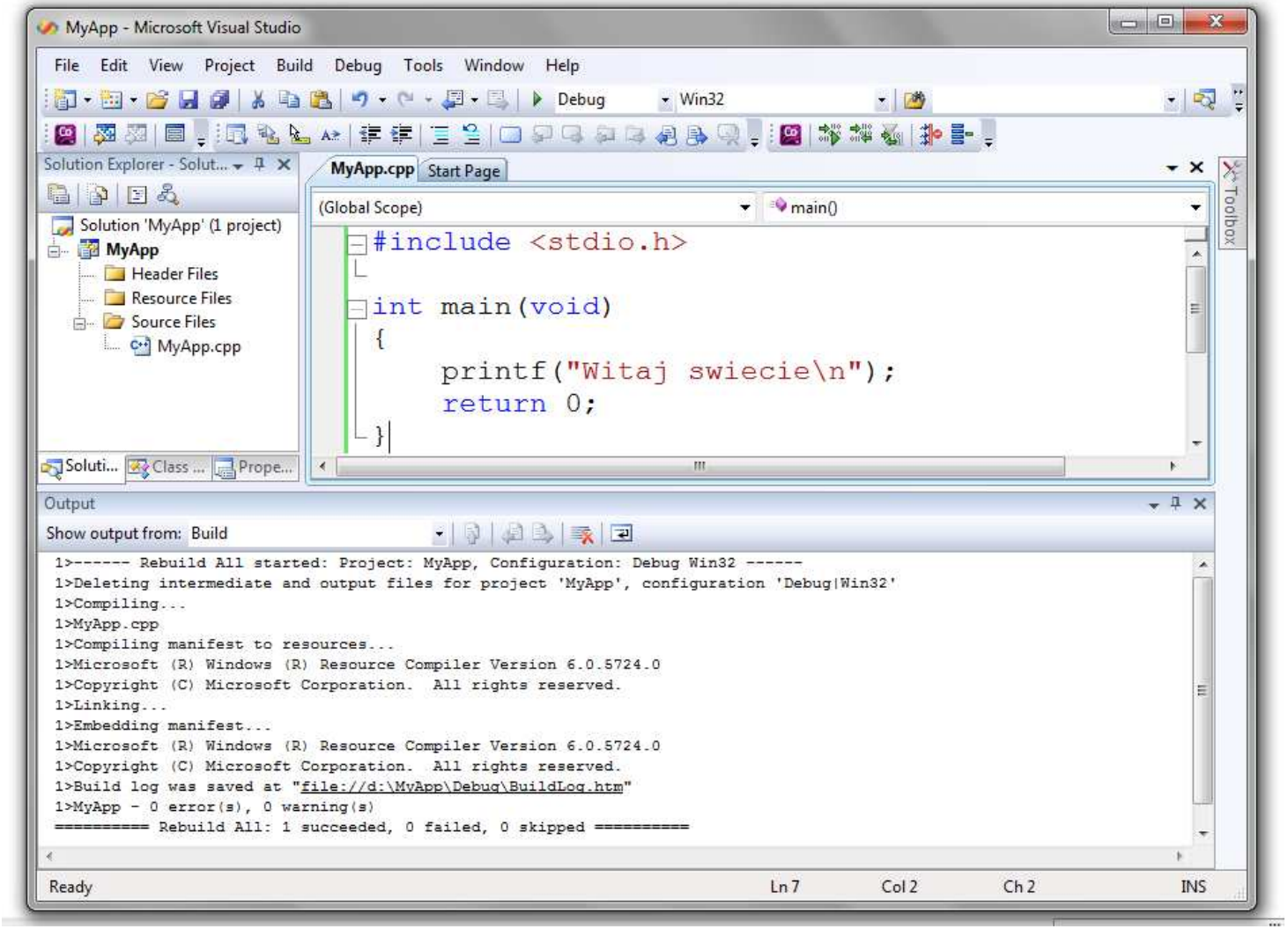
# Język C - pierwszy program

- Wynik uruchomienia programu:

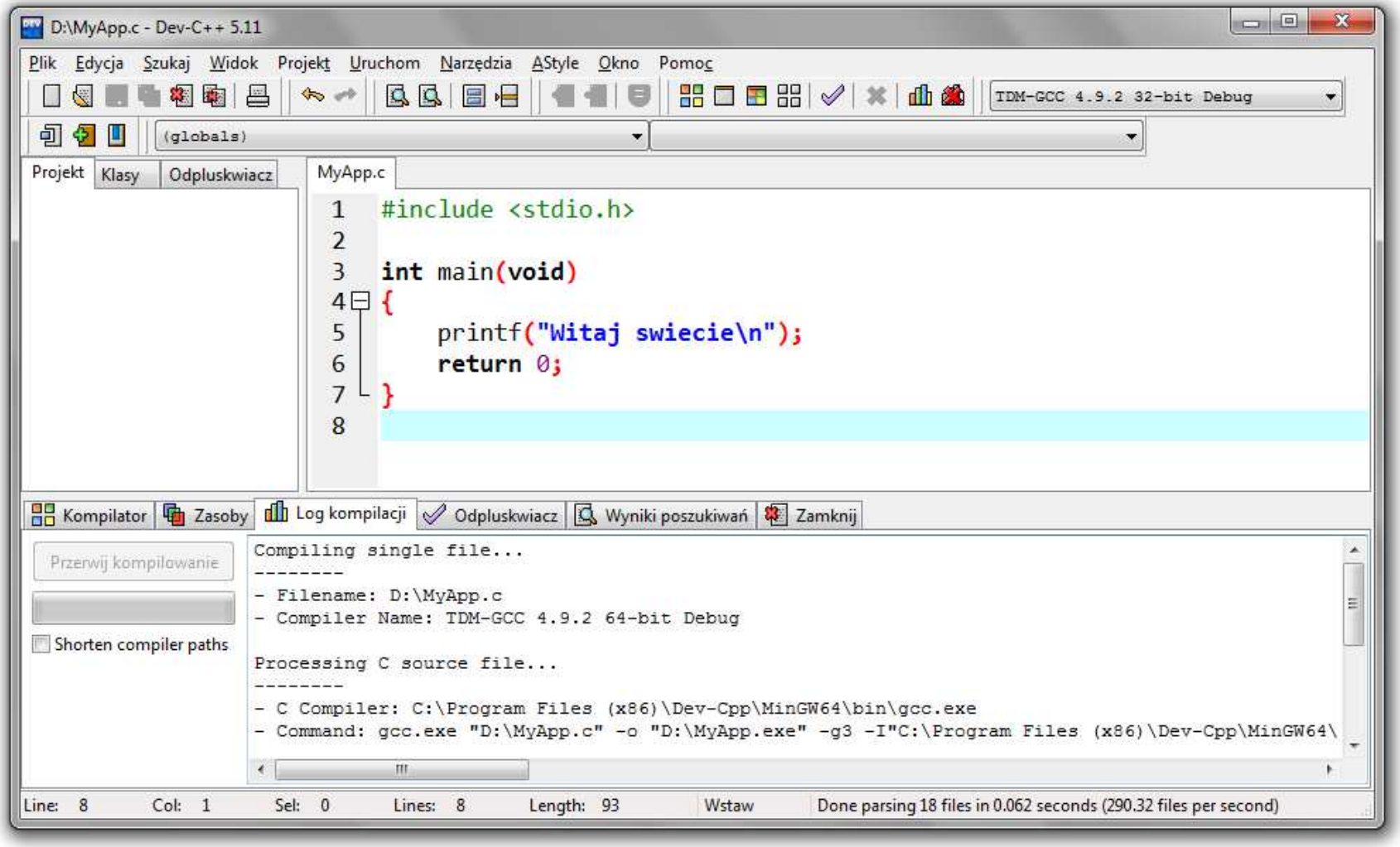
# Język C - struktura programu



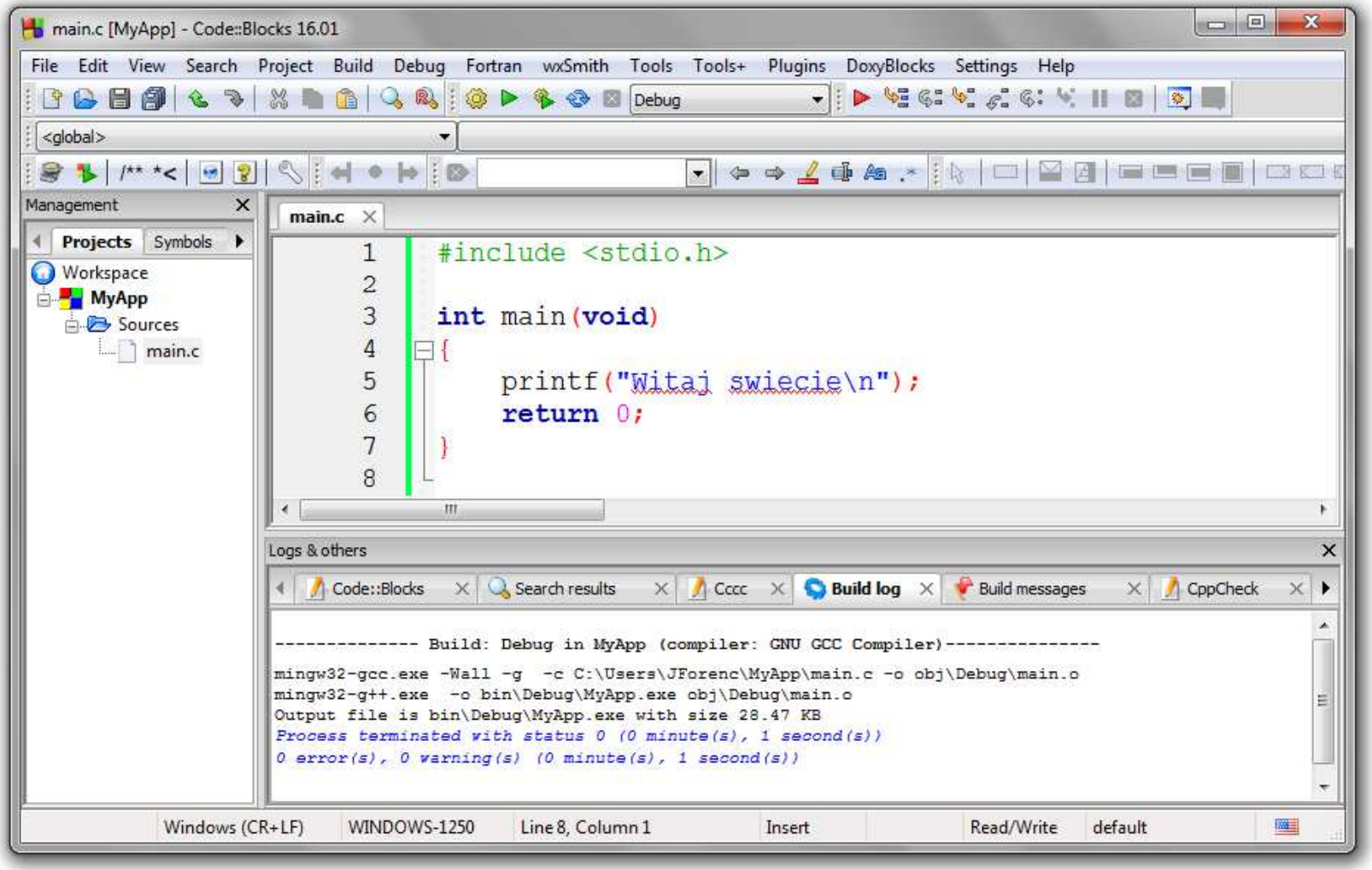
# Microsoft Visual Studio 2008



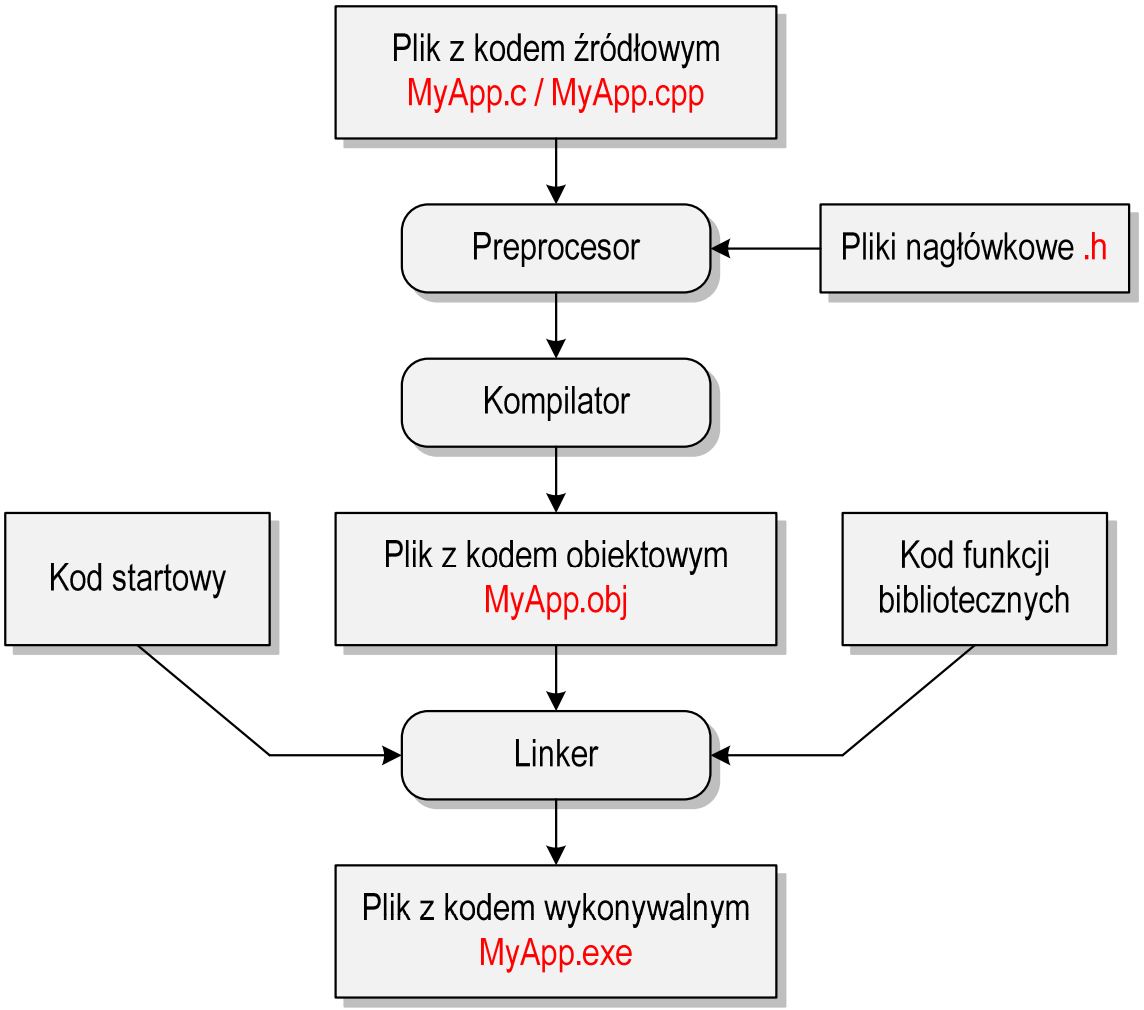
# Dev-C++ 5.11


D:\MyApp.c - Dev-C++ 5.11
Plik Edycja Szukaj Widok Projekt Uruchom Narzędzia AStyle Okno Pomoc
TDM-GCC 4.9.2 32-bit Debug
(globals)
Projekt Klasy Odpluskwiacz
MyApp.c
1 #include <stdio.h>
2
3 int main(void)
4 {
5 printf("Witaj swiecie\n");
6 return 0;
7 }
8
Kompilator Zasoby Log kompilacji Odpluskwiacz Wyniki poszukiwań Zamknij
Przerwij kompilowanie
Shorten compiler paths
Compiling single file...
--------
- Filename: D:\MyApp.c
- Compiler Name: TDM-GCC 4.9.2 64-bit Debug
Processing C source file...
--------
- C Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\gcc.exe
- Command: gcc.exe "D:\MyApp.c" -o "D:\MyApp.exe" -g3 -I"C:\Program Files (x86)\Dev-Cpp\MinGW64\
Line: 8 Col: 1 Sel: 0 Lines: 8 Length: 93 Wstaw
Done parsing 18 files in 0.062 seconds (290.32 files per second)

# Code::Blocks 16.01



# Język C - kompilacja programu



# Język C - zapis kodu programu

- Sposób zapisu kodu programu wpływa tylko na jego przejrzystość, a nie na kompilację i wykonanie
- W takiej postaci program także skompiluje się:

```
#include <stdio.h>
int main(void){printf("Witaj swiecie\n");return 0;}
```

- W Microsoft Visual Studio 2008 można automatycznie sformatować kod źródłowy programu - Ctrl + K + F
- Język C rozróżnia wielkość liter - poniższy kod nie skompiluje się:

```
#include <stdio.h>
int Main(void){printf("Witaj swiecie\n");return 0;}
```

# Język C - Wyświetlanie tekstu (printf)

- Znak przejścia do nowego wiersza \n może pojawić w dowolnym miejscu łańcucha znaków

```
printf("Witaj swiecie\n");
```

```
Witaj swiecie
_
```

```
printf("Witaj\nswiecie\n");
```

```
Witaj
swiecie
_
```

```
printf("Witaj ");
printf("swiecie");
printf("\n");
```

```
Witaj swiecie
_
```

# Język C - Sekwencje sterujące

- Istnieją także inne sekwencje sterujące (ang. escape sequence)

| Opis znaku | Zapis w printf() |
|---|---|
| Alarm (ang. alert), głośniczek wydaje dźwięk | `\a` |
| Backspace | `\b` |
| Wysunięcie strony (ang. form feed) | `\f` |
| Przejście do nowego wiersza (ang. new line) | `\n` |
| CR - Carriage Return (powrót na początek wiersza) | `\r` |
| Tabulacja pozioma (odstęp) (ang. horizontal tab) | `\t` |
| Tabulacja pionowa (ang. vertical tab) | `\v` |

# Język C - Wyświetlenie znaków specjalnych

- Niektóre znaki pełnią specjalną funkcję i nie można wyświetlić ich w tradycyjny sposób

| Opis znaku | Znak | Zapis w printf() |
|---|---|---|
| Cudzysłów | " | \" |
| Apostrof | ' | \' |
| Ukośnik (ang. backslash) | \ | \\ |
| Procent | % | %% |

```
Sciezka dostepu: "C:\dane\plik.txt"
```

```
printf("Sciezka dostepu: \"C:\\dane\\plik.txt\"\n");
```

# Język C - Wyświetlenie znaku o podanym kodzie

- Można wyświetlić dowolny znak podając jego kod w systemie ósemkowym lub szesnastkowym

| Znaczenie | Zapis |
|---|---|
| Znak o podanym kodzie ASCII (system ósemkowy) | `\0oo` |
| Znak o podanym kodzie ASCII (system szesnastkowy) | `\xhh` |

```
printf("\127\151\164\141\152\040");
printf("\x73\x77\x69\x65\x63\x69\x65\x21\x0A");
```

```
Witaj swiecie!
```

# Język C - Wyświetlenie tekstu

```c
#include <stdio.h>

int main(void)
{
    printf("------------------\n");
    printf("| Punkty | Ocena |\n");
    printf("------------------\n");
    printf("| 91-100 |  5,0  |\n");
    printf("|  81-90 |  4,5  |\n");
    printf("|  71-80 |  4,0  |\n");
    printf("|  61-70 |  3,5  |\n");
    printf("|  51-60 |  3,0  |\n");
    printf("|   0-50 |  2,0  |\n");
    printf("------------------\n");

    return 0;
}
```

```
------------------
| Punkty | Ocena |
------------------
| 91-100 |  5,0  |
|  81-90 |  4,5  |
|  71-80 |  4,0  |
|  61-70 |  3,5  |
|  51-60 |  3,0  |
|   0-50 |  2,0  |
------------------
```

# Język C - Komentarze

- Komentarze są pomijane podczas kompilacji

```
/*
  Nazwa: MyApp.cpp
  Autor: Jarosław Forenc, Politechnika Białostocka
  Data:  25-02-2020 12:15
  Opis:  Program wyświetlający tekst "Witaj swiecie"
*/

#include <stdio.h>     // zawiera deklarację printf()

int main(void)         // nagłówek funkcji main()
{
    printf/* funkcja */("Witaj swiecie\n");

    return 0;
}
```

# Informatyka

- Informatyka (ang. computer science)
  - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem informacji
  - w języku polskim termin informatyka zaproponował w październiku 1968 r. prof. Romuald Marczyński na konferencji poświęconej „maszynom matematycznym”
  - wzorem nazwy były francuskie informatique i niemieckie Informatik

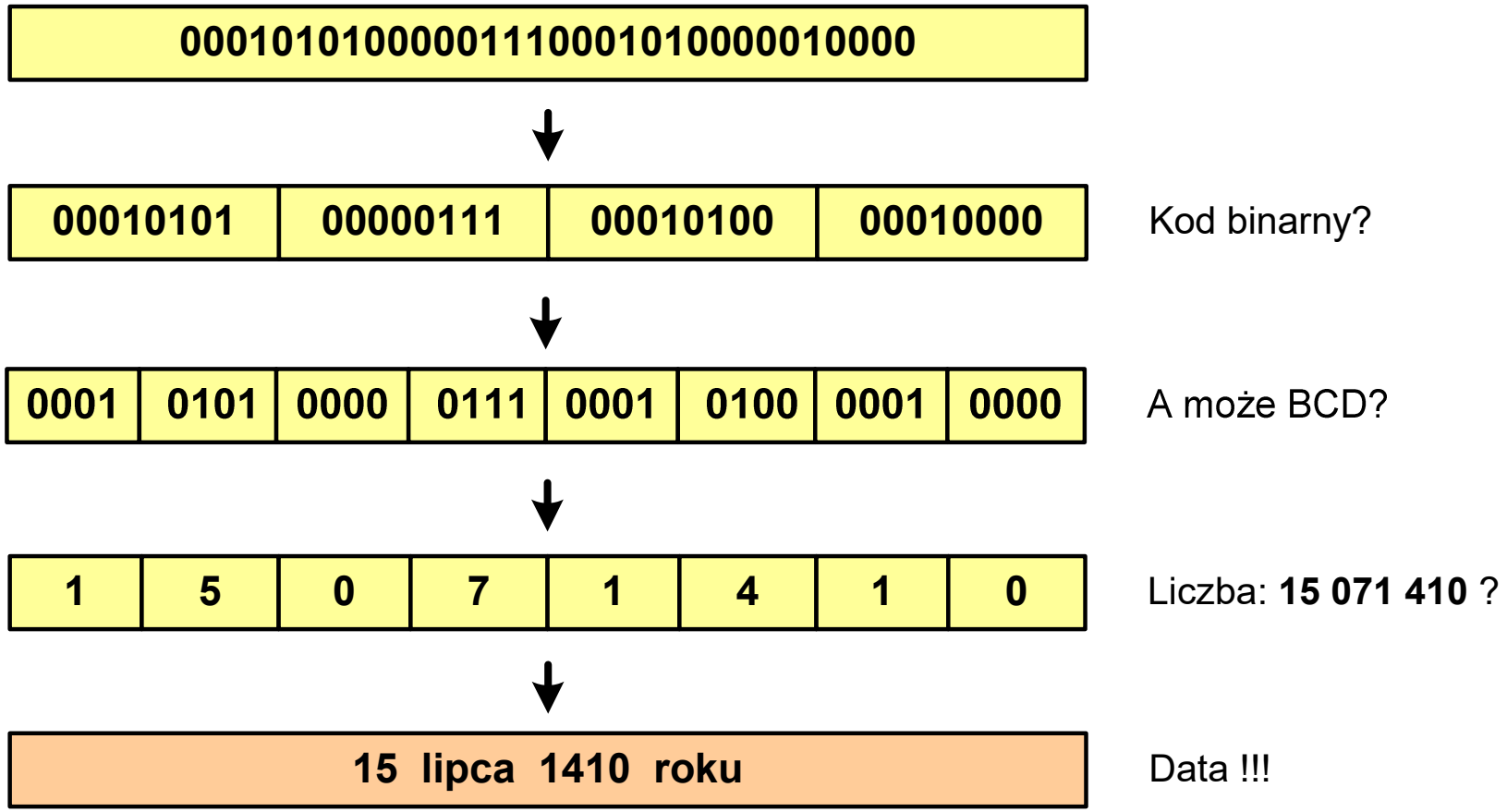
- Informatykę można rozpatrywać jako:
  - samodzielną dyscyplinę naukową
  - narzędzie wykorzystywane przez inne nauki
  - gałąź techniki
  - przemysł wytwarzający sprzęt (hardware) i oprogramowanie (software)

# Informacja

- Informatyka (ang. computer science)
  - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem informacji
- Informacja - wielkość abstrakcyjna, która może być:
  - przechowywana w pewnych obiektach
  - przesyłana pomiędzy pewnymi obiektami
  - przetwarzana w pewnych obiektach
  - stosowana do sterowania pewnymi obiektami
- Dane - surowe fakty i liczby
- Przetwarzanie danych - logicznie powiązany zespół czynności pozwalających na uzyskanie z danych niezbędnych informacji

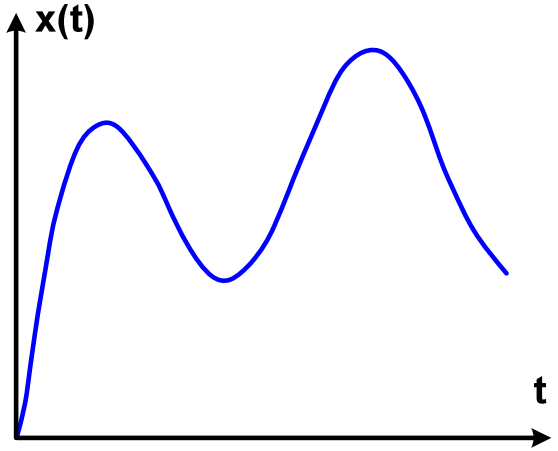
# Informacja

- Co oznaczają poniższe dane?

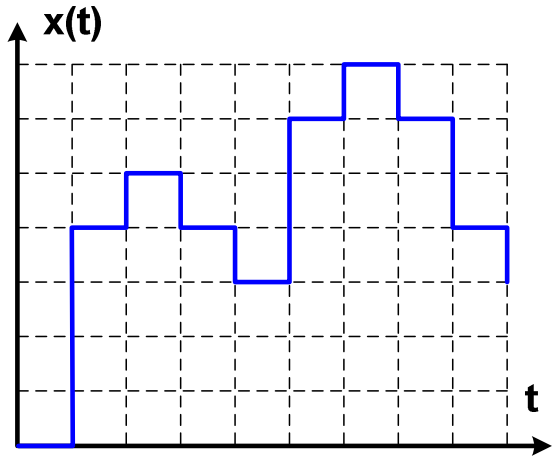


# Informacja analogowa i cyfrowa

- Sygnał analogowy
  - może przyjmować dowolną wartość z ciągłego przedziału (nieskończonego lub ograniczonego zakresem zmienności)
  - wartości mogą zostać określone w każdej chwili czasu dzięki funkcji matematycznej opisującej dany sygnał




- Sygnał cyfrowy
  - dziedzina i zbiór wartości są dyskretne
  - sygnał ciągły, który może zmieniać swoją wartość tylko w określonych chwilach czasu i może przyjmować tylko określone wartości



# Informacja analogowa i cyfrowa

- Zalety sygnałów cyfrowych:
  - odporne na zakłócenia
  - powtarzalne (np. kopia filmu na DVD i VHS)
  - możliwość przesyłania na duże odległości
  - możliwość szyfrowania sygnału (kryptografia)
  - niższe koszty przetwarzania
- Wady sygnałów cyfrowych:
  - ograniczenie częstotliwości próbkowania (sygnał analogowy zamieniony na cyfrowy i ponownie na analogowy nie jest już tym samym sygnałem)

# Liczby i cyfry

- Liczba - pojęcie abstrakcyjne, abstrakcyjny wynik obliczeń, wartość
  - umożliwia wyrażenie wyniku liczenia przedmiotów oraz mierzenia wielkości
- Cyfra - umowny znak (symbol) stosowany do zapisu liczby
  - liczba znaków służących do zapisu jest zależna od systemu liczbowego i przyjętego sposobu zapisu
  - system dziesiętny - 10 znaków
  - system szesnastkowy - 16 znaków
  - system rzymski - 7 znaków
- Cyfry rzymskie

| I | V | X | L | C | D | M |
|---|---|---|---|---|---|---|
| *1* | *5* | *10* | *50* | *100* | *500* | *1000* |

# Liczby i cyfry

- Cyfry arabskie (pochodzą z Indii)
  - arabskie, standardowe europejskie

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
|---|---|---|---|---|---|---|---|---|---|

  - indyjsko-arabskie

| ١ | ٢ | ٣ | ٤ | ٥ | ٦ | ٧ | ٨ | ٩ | ٠ |
|---|---|---|---|---|---|---|---|---|---|
| *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *0* |

  - wschodnio-indyjsko-arabskie

| ١ | ٢ | ٣ | ۴ | ۵ | ۶ | ٧ | ٨ | ٩ | ٠ |
|---|---|---|---|---|---|---|---|---|---|
| *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *0* |

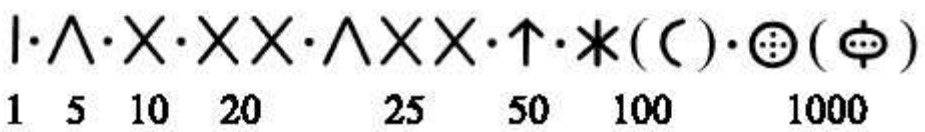
- W niektórych systemach jako cyfry stosowane są litery, np.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* |

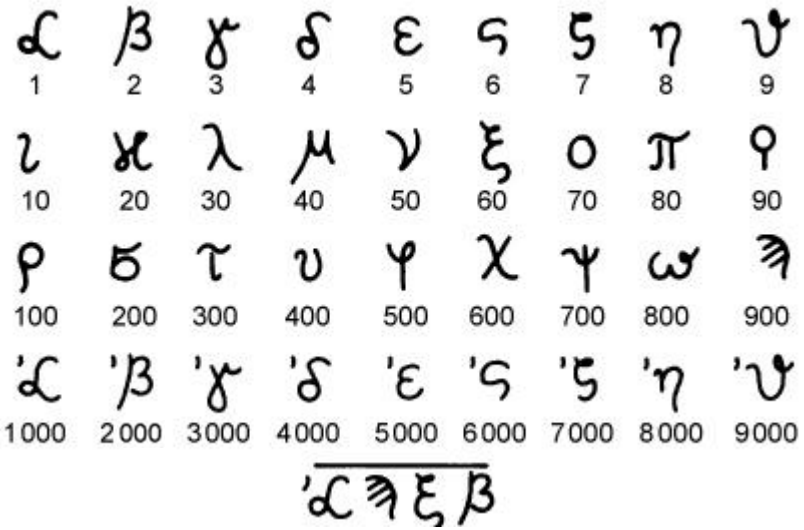
# Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

**cyfry etruskie**




**cyfry grecko-jońskie**

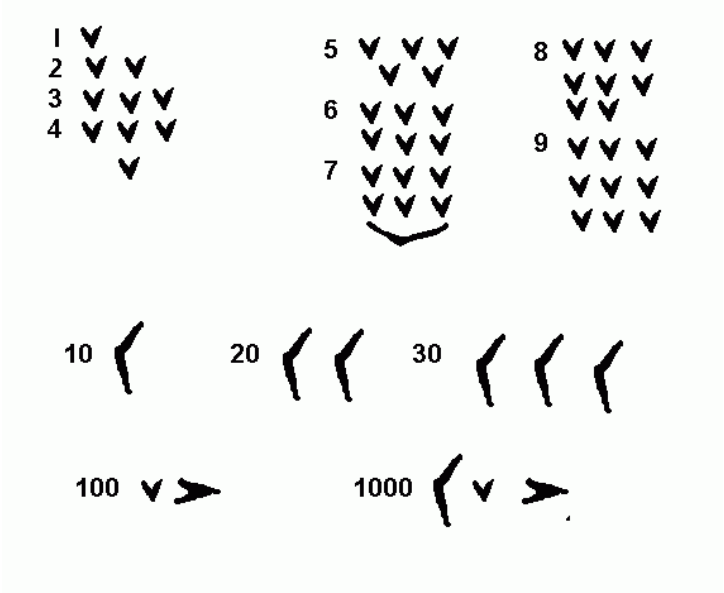



**cyfry w pisowni chińskiej**

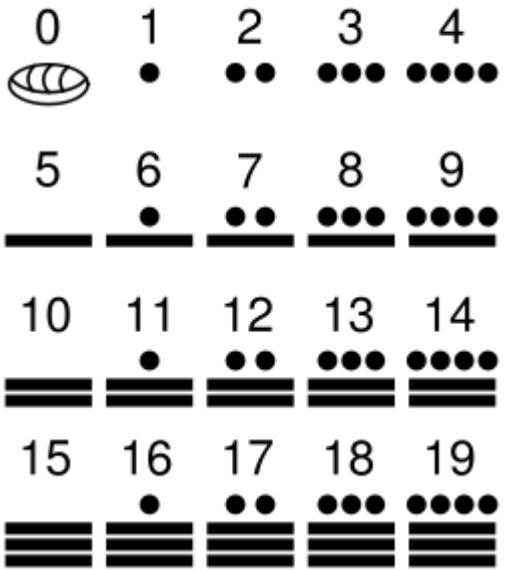
# Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

**liczby w piśmie klinowym (Babilończycy)**

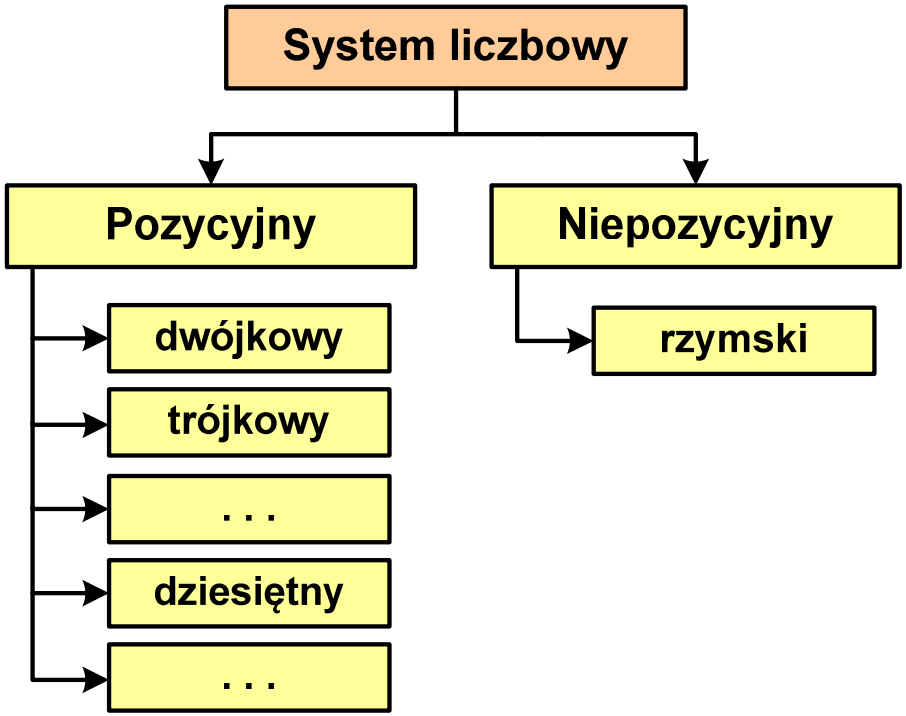



**system prekolumbijski**



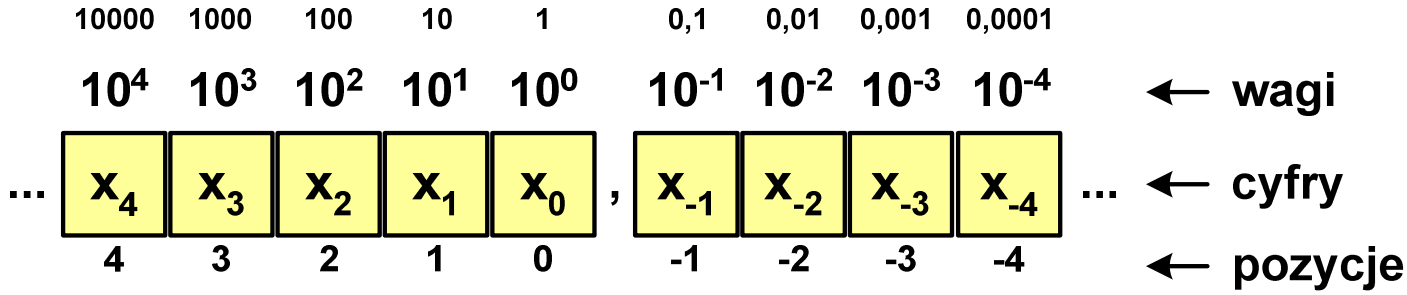
# Systemy liczbowe

- System liczbowy - zbiór zasad umożliwiających zapis liczb za pomocą cyfr oraz wykonywanie działań na tych liczbach

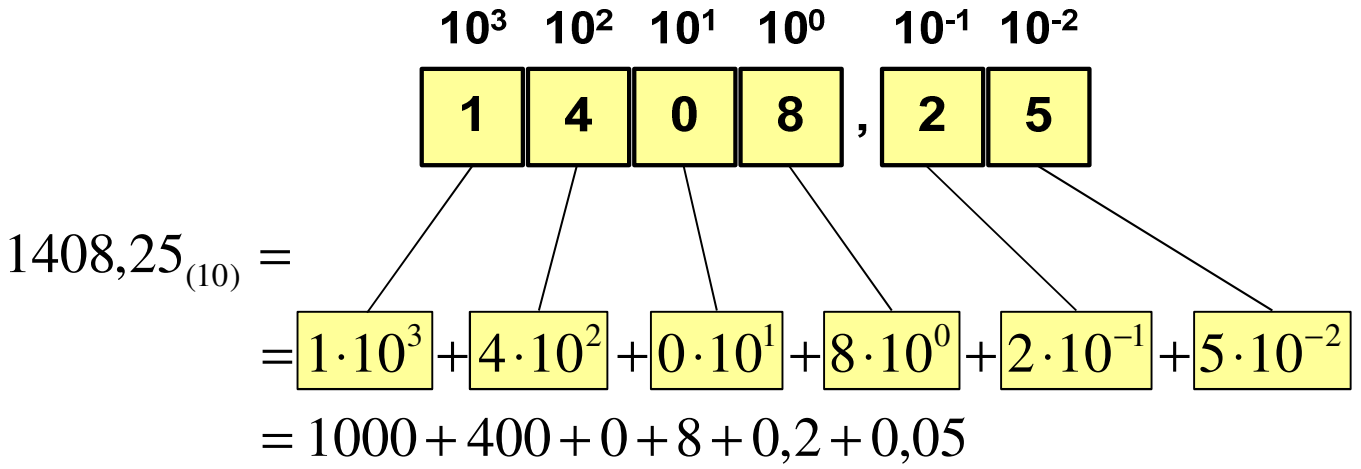



- Pozycyjny - znaczenie cyfry jest zależne od miejsca (pozycji), które zajmuje ona w liczbie
  - system dziesiętny - liczba 111 (każda cyfra ma inne znaczenie)
- Niepozycyjny - znaczenie cyfry jest niezależne od miejsca położenia w liczbie
  - system rzymski - liczba III

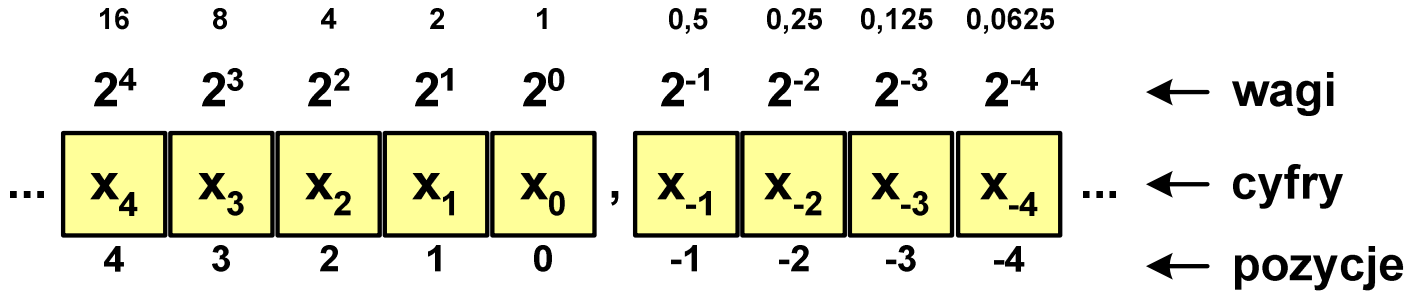
# System dziesiętny (ang. decimal)



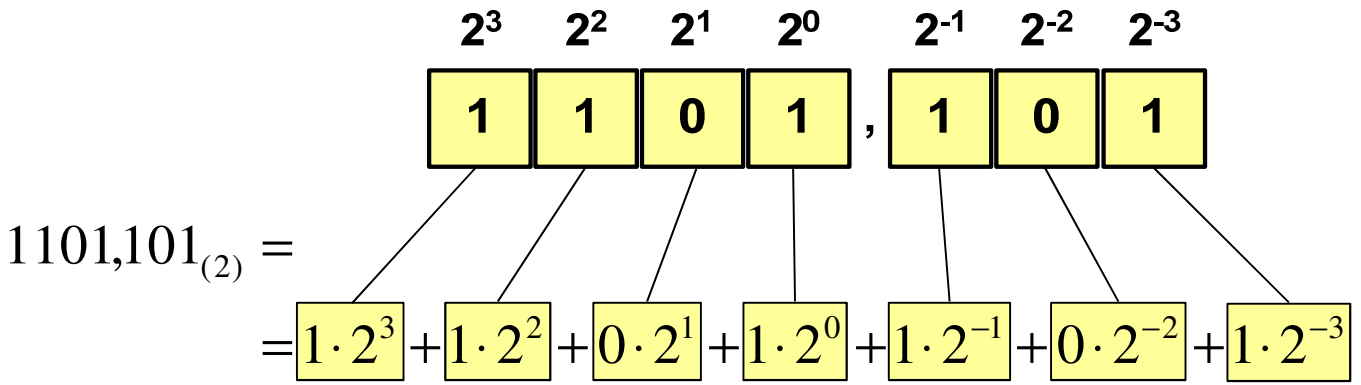

- p - podstawa systemu pozycyjnego, D - zbiór dozwolonych cyfr
- p = 10, D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}



# System dwójkowy (ang. binary)



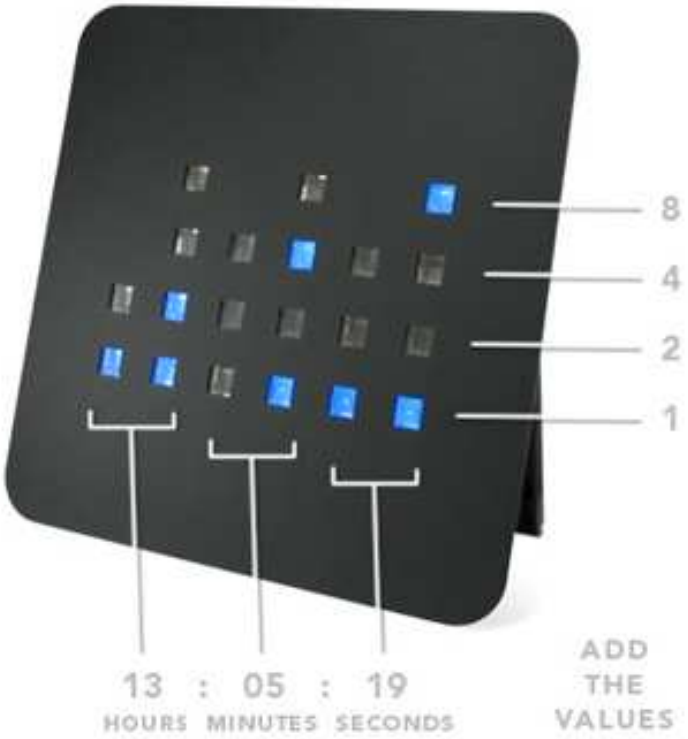

- w systemie dwójkowym: p = 2, D = {0, 1}




$$1101{,}101_{(2)} =$$
$$= 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3}$$
$$= 8 + 4 + 0 + 1 + 0{,}5 + 0 + 0{,}125$$
$$= 13{,}625_{(10)}$$

# System dwójkowy - zastosowania

- Powszechnie używany w informatyce, technice cyfrowej



# System szesnastkowy (ang. hexadecimal)

- System heksadecymalny
- p = 16, D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
- Powszechnie używany w informatyce - jeden bajt można zapisać za pomocą tylko dwóch cyfr szesnastkowych

$$3A5D_{(16)} = 3 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16^1 + 13 \cdot 16^0 = 14941_{(10)}$$

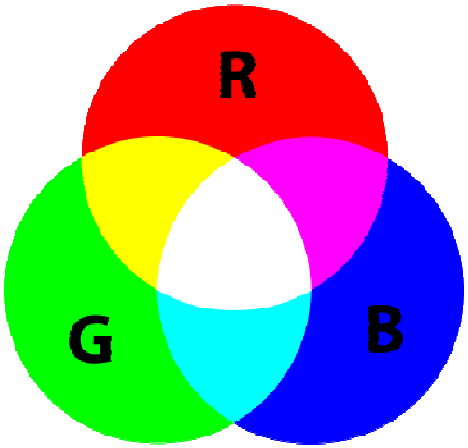
- Sposoby zapisu liczb w systemie szesnastkowym:

| | | |
|---|---|---|
| 3A5Dh | 0x3A5D | #3A5D |
| $3A5D_{(16)}$ | $3A5D_{16}$ | $3A5D_{hex}$ |
| $(3A5D)_{hex}$ | $(3A5D)_{16}$ | $3A5D |

# System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Każda barwa przyjmuje wartość z zakresu: $0..255_{(10)}$, $00..FF_{(16)}$

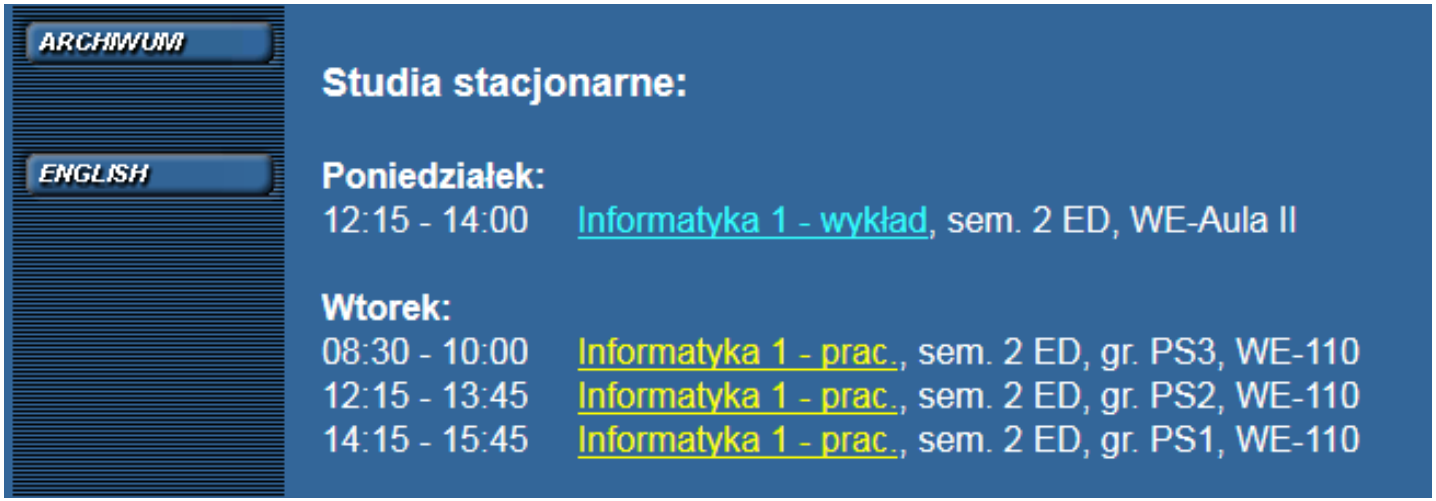
#FF48B8



# System szesnastkowy - zastosowania

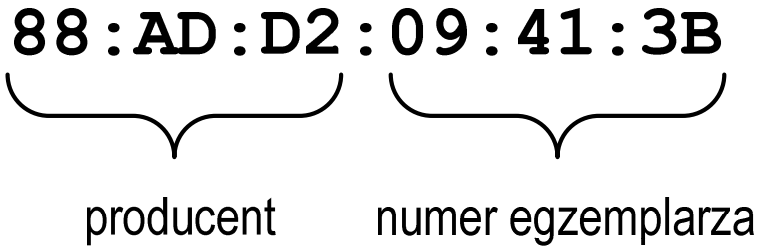
- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Kolory w dokumentach HTML:

```
<BODY bgcolor="#336699" text="#000000" link="#FFFF00"
 vlink="#33FFFF" alink="#FF0000">
```



# System szesnastkowy - zastosowania

- 48-bitowy adres fizyczny urządzenia (MAC - Media Access Control)




- http://hwaddress.com

# Przykład systemu niepozycyjnego - system rzymski

- W systemie rzymskim posługujemy się siedmioma znakami:

  I - 1  V - 5  X - 10  L - 50  C - 100  D - 500  M - 1000

- Za pomocą dostępnych symboli można określić liczby od 1 do 3999
- System addytywny - wartość liczby określa się na podstawie sumy wartości cyfr, np.
  - II (1 + 1 = 2), XXX (10 + 10 + 10 = 30)
  - CLX (100 + 50 + 10 = 160), MMXII (1000 + 1000 + 10 + 1 + 1 = 2012)
- Wyjątkiem od powyższej zasady są liczby do opisu których używa się odejmowania, np.
  - IV (5 - 1 = 4), IX (10 - 1 = 9), XL (50 - 10 = 40), XC (100 - 10 = 90)
- Stosowany w łacińskiej części Europy do końca Średniowiecza
- Niewygodny w prowadzeniu nawet prostych działań arytmetycznych, brak ułamków

# Przykład systemu niepozycyjnego - system rzymski

- Zasady tworzenia liczb:
  - zestawiamy odpowiednie znaki od oznaczającego liczbę największą do oznaczającego liczbę najmniejszą

    XVI = 10(X) + 5(V) + 1(I) = 16

  - jeżeli składnik liczby, którą piszemy, jest wielokrotnością liczby nominalnej, wtedy zapisywany jest z użyciem kilku następujących po sobie znaków

    CCC = 100(C) + 100(C) + 100(C) = 300

  - dodatkowo należy zachować zasadę nie pisania czterech tych samych znaków po sobie, lecz napisać jeden znak wraz ze znakiem oznaczającym wartość większą o jeden rząd liczbowy

    CD = 500(D) - 100(C) = 400

# Przykład systemu niepozycyjnego - system rzymski

- Zasady odczytu liczb:
  - cyfry jednakowe są dodawane

    MMM = 1000(M) + 1000(M) + 1000(M) = 3000
  - cyfry mniejsze stojące przed większymi są odejmowane od nich

    CDXCIV = 500(D) - 100(C) + 100(C) - 10(X) + 5(V) - 1(I) = 494
  - cyfry mniejsze stojące za większymi są do nich dodawane

    MDCLX = 1000(M) + 500(D) + 100(C) + 50(L) + 10 (X) = 1660

# Konwersja na system dziesiętny

- p = 4, D = {0, 1, 2, 3}

| $4^4$ | $4^3$ | $4^2$ | $4^1$ | $4^0$ |
|---|---|---|---|---|
| 2 | 1 | 3 | 0 | 2 |

$$21302_{(4)} = ?_{(10)}$$
$$21302_{(4)} = 2\cdot 4^0 + 0\cdot 4^1 + 3\cdot 4^2 + 1\cdot 4^3 + 2\cdot 4^4$$
$$21302_{(4)} = 2\cdot 1 + 0\cdot 4 + 3\cdot 16 + 1\cdot 64 + 2\cdot 256$$
$$21302_{(4)} = 2 + 0 + 48 + 64 + 512 = 626_{(10)}$$

- p = 17, D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G}

| $17^3$ | $17^2$ | $17^1$ | $17^0$ |
|---|---|---|---|
| A | C | 2 | 4 |

$$AC24_{(17)} = ?_{(10)}$$
$$AC24_{(17)} = 4\cdot 17^0 + 2\cdot 17^1 + 12\cdot 17^2 + 10\cdot 17^3$$
$$AC24_{(17)} = 4\cdot 1 + 2\cdot 17 + 12\cdot 289 + 10\cdot 4913$$
$$AC24_{(17)} = 4 + 34 + 3468 + 49130 = 52636_{(10)}$$

# Konwersja na system dziesiętny (schemat Hornera)

- p = 4, D = {0, 1, 2, 3}

$$21302_{(4)} = w_{(10)} \qquad x_4\, x_3\, x_2\, x_1\, x_0 = w_{(10)}$$

$$w_{(10)} = 0$$

$$w_{(10)} = x_4 + w_{(10)} \cdot p = 2 + 0 \cdot 4 = 2$$

$$w_{(10)} = x_3 + w_{(10)} \cdot p = 1 + 2 \cdot 4 = 9$$

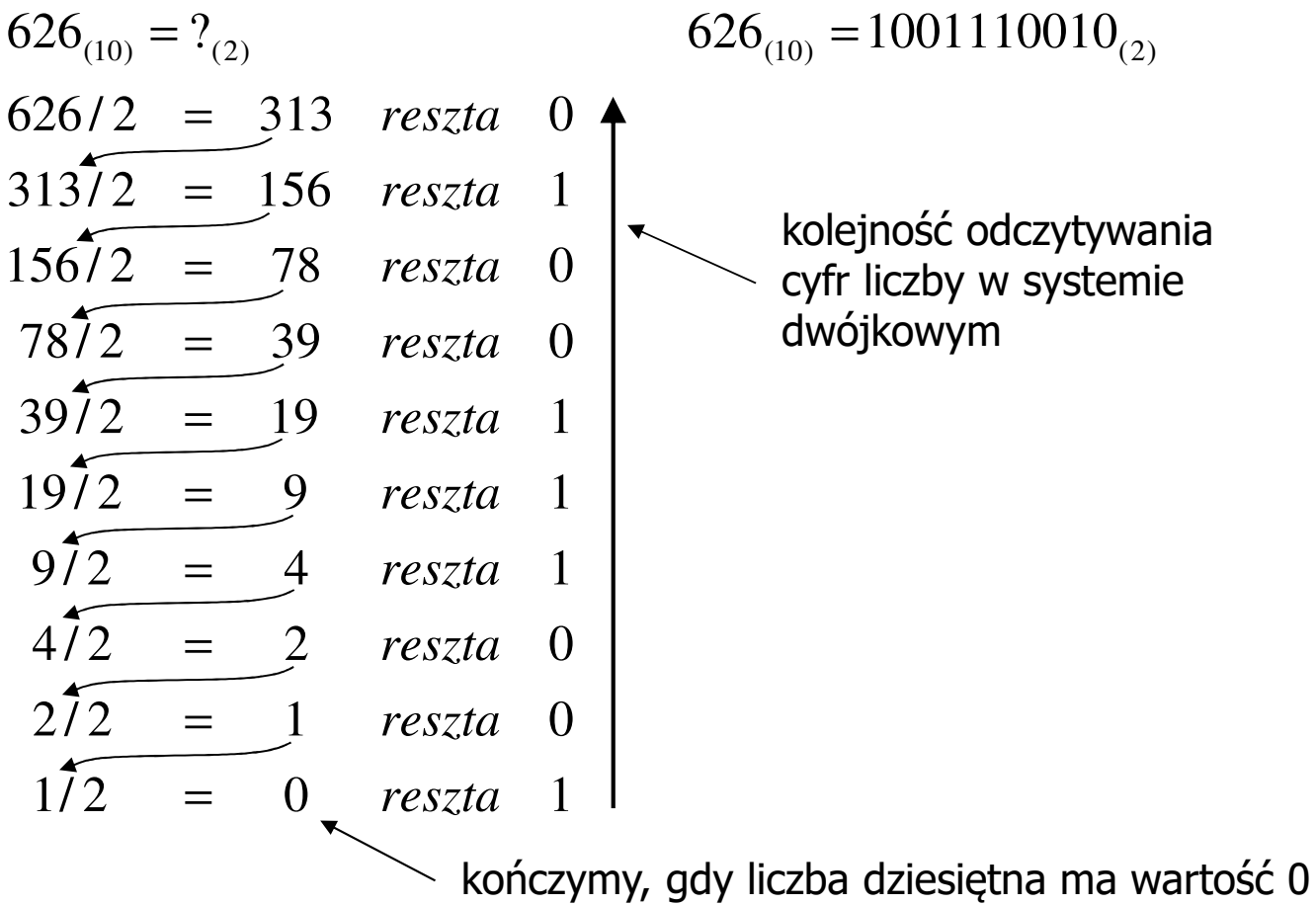
$$w_{(10)} = x_2 + w_{(10)} \cdot p = 3 + 9 \cdot 4 = 39$$

$$w_{(10)} = x_1 + w_{(10)} \cdot p = 0 + 39 \cdot 4 = 156$$

$$w_{(10)} = x_0 + w_{(10)} \cdot p = 2 + 156 \cdot 4 = 626_{(10)}$$

# Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu p = 10 na system p = 2



# Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu p = 10 na system p = 7

$626_{(10)} = ?_{(7)}$ $\qquad$ $626_{(10)} = 1553_{(7)}$

$$
\begin{aligned}
626/7 &= 89 \quad \textit{reszta} \quad 3 \\
89/7 &= 12 \quad \textit{reszta} \quad 5 \\
12/7 &= 1 \quad \textit{reszta} \quad 5 \\
1/7 &= 0 \quad \textit{reszta} \quad 1
\end{aligned}
$$

- zamiana liczby z systemu p = 10 na system p = 14

$626_{(10)} = ?_{(14)}$ $\qquad$ $626_{(10)} = 32\mathrm{A}_{(14)}$

$$
\begin{aligned}
626/14 &= 44 \quad \textit{reszta} \quad 10 \rightarrow \mathrm{A} \\
44/14 &= 3 \quad \textit{reszta} \quad 2 \\
3/14 &= 0 \quad \textit{reszta} \quad 3
\end{aligned}
$$

# Szybkie konwersje: 2 → 4, 8, 16 4, 8, 16 → 2

## 2 → 4

$$110110011_{(2)} = ?_{(4)}$$

$$\underbrace{01}_{1} \mid \underbrace{10}_{2} \mid \underbrace{11}_{3} \mid \underbrace{00}_{0} \mid \underbrace{11}_{3}$$

$$110110011_{(2)} = 12303_{(4)}$$

## 2 → 8

$$10110011_{(2)} = ?_{(8)}$$

$$\underbrace{010}_{2} \mid \underbrace{110}_{6} \mid \underbrace{011}_{3}$$

$$10110011_{(2)} = 263_{(8)}$$

## 2 → 16

$$1011010_{(2)} = ?_{(16)}$$

$$\underbrace{0101}_{5} \mid \underbrace{1010}_{A}$$

$$1011010_{(2)} = 5A_{(16)}$$

## 4 → 2

$$12303_{(4)} = ?_{(2)}$$

$$\overbrace{01}^{1} \quad \overbrace{10}^{2} \quad \overbrace{11}^{3} \quad \overbrace{00}^{0} \quad \overbrace{11}^{3}$$

$$12303_{(4)} = 110110011_{(2)}$$

## 8 → 2

$$263_{(8)} = ?_{(2)}$$

$$\overbrace{010}^{2} \quad \overbrace{110}^{6} \quad \overbrace{011}^{3}$$

$$263_{(8)} = 10110011_{(2)}$$

## 16 → 2

$$5A_{(16)} = ?_{(2)}$$

$$\overbrace{0101}^{5} \quad \overbrace{1010}^{A}$$

$$5A_{(16)} = 1011010_{(2)}$$

# Koniec wykładu nr 1

Dziękuję za uwagę!