

Informatyka 1 (ES1E2009)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr II, studia stacjonarne I stopnia
Rok akademicki 2020/2021

Wykład nr 3 (08.03.2021)

dr inż. Jarosław Forenc

Plan wykładu nr 3

- Język C
 - operator warunkowy
 - instrukcja switch
 - pętla for
 - operatory ++ i --
- 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 - Operator warunkowy

- Operator warunkowy składa się z dwóch symboli i trzech operandów

```
wyrażenie1 ? wyrażenie2 : wyrażenie3
```

- Najczęściej zastępuje proste instrukcje **if-else**

```
float akcyza, cena, pojemnosc;
```

```
if (pojemnosc <= 2000)
    akcyza = cena*0.031;    /* 3.1% */
else
    akcyza = cena*0.186;    /* 18.6% */
```

```
akcyza = pojemnosc <= 2000 ? cena*0.031 : cena*0.186 ;
```

Język C - Operator warunkowy

```
if (x < 0)
    y = -x;
else
    y = x;
```

```
y = x < 0 ? -x : x;
```

- obliczenie modułu liczby x

```
if (a > b)
    max = a;
else
    max = b;
```

```
max = a > b ? a : b;
```

- wyznaczenie max z dwóch liczb

- Operator warunkowy ma bardzo niski priorytet
- Niższy priorytet mają tylko operatory przypisania (=, +=, -=, ...) i operator przecinkowy (,)

Przykład: operator warunkowy

- Studenci chcą dojechać z akademika do sklepu - ile taksówek powinni zamówić? (Jedna taksówka może przewieźć 4 osoby.)

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int st, taxi;
```

```
    printf("Podaj liczbe studentow: ");
```

```
    scanf("%d", &st);
```

```
    taxi = st / 4 + (st % 4 ? 1 : 0);
```

```
    printf("Liczba taxi: %d\n", taxi);
```

```
    return 0;
```

```
}
```

```
Podaj liczbe studentow: 23  
Liczba taxi: 6
```

Przykład: sprawdzenie parzystości liczby

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int x;
```

```
    printf("Podaj x: ");
```

```
    scanf("%d", &x);
```

```
    if (x%2==0)
```

```
        printf("Liczba parzysta\n");
```

```
    else
```

```
        printf("Liczba nieparzysta\n");
```

```
    printf("Liczba %s\n", x%2==0 ? "parzysta": "nieparzysta");
```

```
    return 0;
```

```
}
```

Podaj x: -3

Liczba nieparzysta

Liczba nieparzysta

Język C - Instrukcja switch

- Instrukcja wyboru wielowariantowego **switch**

```
switch (wyrażenie)
{
    case wyrażenie Stałe: instrukcje;
    case wyrażenie Stałe: instrukcje;
    case wyrażenie Stałe: instrukcje;
    ...
    default: instrukcje;
}
```

- **wyrażenie Stałe** - wartość typu całkowitego, znana podczas kompilacji
 - stała liczbowa, np. **3**, **5**, **9**
 - znak w apostrofach, np. **'a'**, **'z'**, **'+'**
 - stała zdefiniowana przez **const** lub **#define**

Język C - Instrukcja switch

- Program wyświetlający słownie liczbę z zakresu 1..5 wprowadzoną z klawiatury

```
#include <stdio.h>

int main(void)
{
    int liczba;

    printf("Podaj liczbę (1..5): ");
    scanf("%d", &liczba);
```


Język C - Instrukcja switch

```
switch (liczba)
{
    case 1: printf("Liczba: jeden\n");
            break;
    case 2: printf("Liczba: dwa\n");
            break;
    case 3: printf("Liczba: trzy\n");
            break;
    case 4: printf("Liczba: cztery\n");
            break;
    case 5: printf("Liczba: piec\n");
            break;
    default: printf("Inna liczba\n");
}
}
```

Podaj liczbe: 2
Liczba: dwa

Podaj liczbe: 0
Inna liczba

Język C - Instrukcja switch

```
switch (liczba)
{
    case 1:
    case 3:
    case 5: printf("Liczba nieparzysta\n");
            break;
    case 2:
    case 4: printf("Liczba parzysta\n");
            break;
    default: printf("Inna liczba\n");
}
```

Podaj liczbe: 2
Liczba parzysta

- Te same instrukcje mogą być wykonane dla kilku etykiet **case**

Język C - Instrukcja switch

```
switch (liczba)
{
    case 1: case 3: case 5:
        printf("Liczba nieparzysta\n");
        break;
    case 2: case 4:
        printf("Liczba parzysta\n");
        break;
    default: printf("Inna liczba\n");
}
```

Podaj liczbe: 2
Liczba parzysta

- Etykiety **case** mogą być pisane w jednym wierszu

Język C - Instrukcja switch

```
switch (liczba%2)
{
    case 1: case -1:
        printf("Liczba nieparzysta\n");
        break;
    case 0:
        printf("Liczba parzysta\n");
}
```

Podaj liczbe: 2
Liczba parzysta

- Część domyślna (**default**) może być pominięta

Język C - Instrukcja switch (bez break)

```
switch (liczba)
{
    case 1: printf("Liczba: jeden\n");
    case 2: printf("Liczba: dwa\n");
    case 3: printf("Liczba: trzy\n");
    case 4: printf("Liczba: cztery\n");
    case 5: printf("Liczba: piec\n");
    default: printf("Inna liczba\n");
}
```

```
Podaj liczbe: 2
Liczba: dwa
Liczba: trzy
Liczba: cztery
Liczba: piec
Inna liczba
```

- Pominięcie instrukcji **break** spowoduje wykonanie wszystkich instrukcji występujących po danym **case** (do końca **switch**)

Przykład: suma kolejnych 10 liczb: $1+2+\dots+10$

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int suma;
```

```
    suma = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;
```

```
    printf("Suma wynosi: %d\n", suma);
```

```
    return 0;
```

```
}
```

Suma wynosi: 55

Przykład: suma kolejnych 100 liczb: $1+2+\dots+100$

```
#include <stdio.h>

int main(void)
{
    int suma=0, i;

    for (i=1; i<=100; i=i+1)
        suma = suma + i;

    printf("Suma wynosi: %d\n", suma);

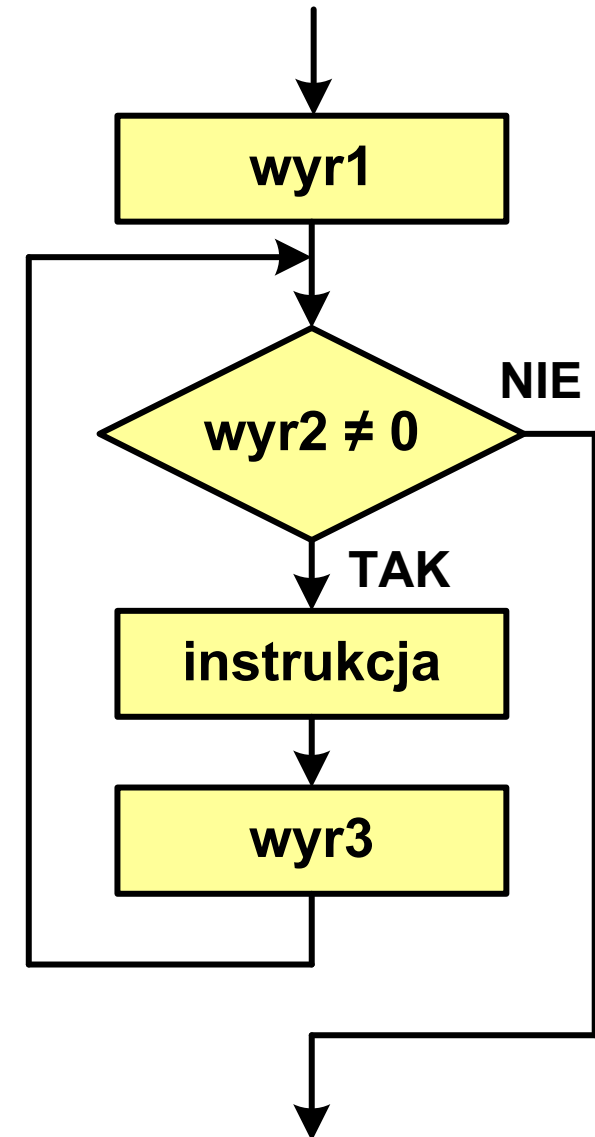
    return 0;
}
```

Suma wynosi: 5050

Język C - pętla for

```
for (wyr1; wyr2; wyr3)  
instrukcja
```

- **wyr1, wyr2, wyr3** - dowolne wyrażenia w języku C
- Instrukcja:
 - **prosta** - jedna instrukcja zakończona średnikiem
 - **złożona** - jedna lub kilka instrukcji objętych nawiasami klamrowymi



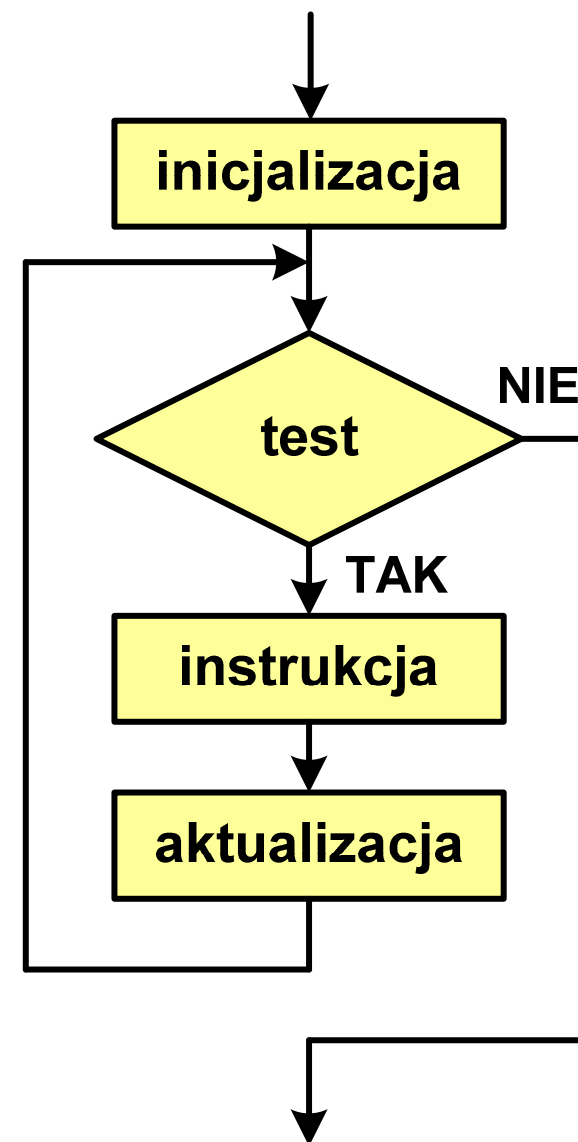
Język C - pętla for

- Najczęściej stosowana postać pętli **for**

```
int i;  
for (i = 0; i < 10; i = i + 1)  
    instrukcja
```

- Instrukcja zostanie wykonana 10 razy (dla $i = 0, 1, 2, \dots, 9$)
- Funkcje pełnione przez wyrażenia

```
for (inicjalizacja; test; aktualizacja)  
    instrukcja
```



Przykład: wyświetlenie tekstu 5 razy

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    for (i=0; i<5; i=i+1)
```

```
        printf("Programowanie nie jest trudne\n");
```

```
    return 0;
```

```
}
```

```
Programowanie nie jest trudne  
Programowanie nie jest trudne  
Programowanie nie jest trudne  
Programowanie nie jest trudne  
Programowanie nie jest trudne
```

Przykład - suma liczb: $1 + 2 + \dots + N$

```
#include <stdio.h>
#define N 1234

int main(void)
{
    int i, suma=0;

    for (i=1; i<=N; i++)
        suma = suma + i;

    printf("Suma %d liczb to %d\n", N, suma);

    return 0;
}
```

Suma 1234 liczb to 761995

Język C - pętla for (przykłady)

```
for (i=0; i<10; i++)  
    printf("%d ", i);
```

0 1 2 3 4 5 6 7 8 9

```
for (i=0; i<10; i++)  
    printf("%d ", i+1);
```

1 2 3 4 5 6 7 8 9 10

```
for (i=1; i<=10; i++)  
    printf("%d ", i);
```

1 2 3 4 5 6 7 8 9 10

Język C - pętla for (przykłady)

```
for (i=1; i<10; i=i+2)  
    printf("%d ", i);
```

1 3 5 7 9

```
for (i=10; i>0; i--)  
    printf("%d ", i);
```

10 9 8 7 6 5 4 3 2 1

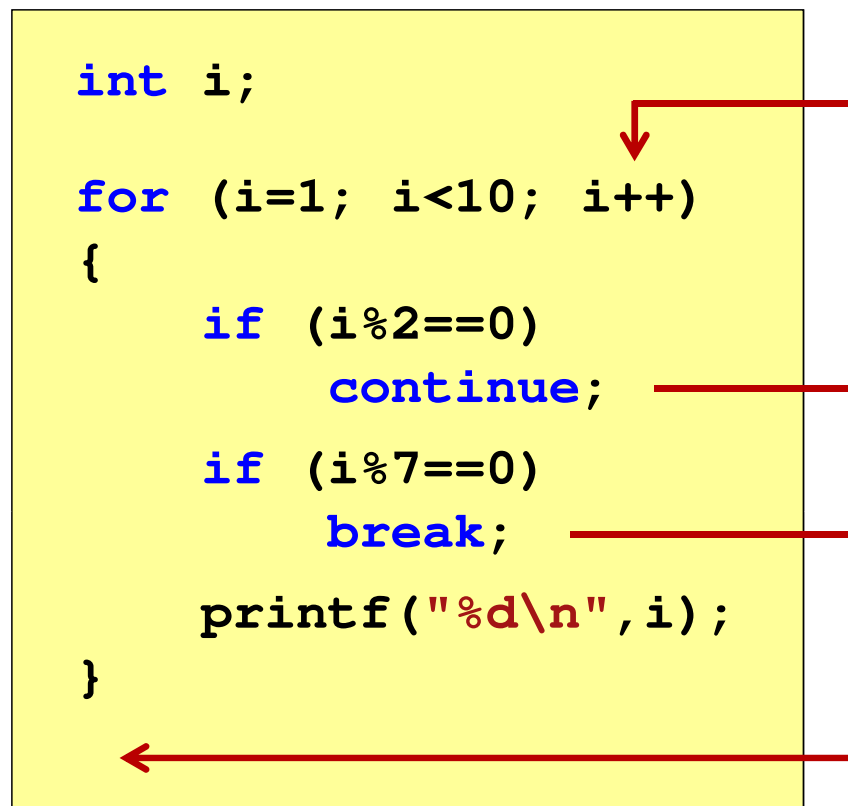
```
for (i=-9; i<=9; i=i+3)  
    printf("%d ", i);
```

-9 -6 -3 0 3 6 9

Język C - pętla for (break, continue)

- W pętli **for** można stosować instrukcje skoku: **break** i **continue**

```
int i;
for (i=1; i<10; i++)
{
    if (i%2==0)
        continue;
    if (i%7==0)
        break;
    printf("%d\n", i);
}
```



1 3 5

- **continue** przerywa bieżącą iterację i przechodzi do obliczania **wyr3**
- **break** przerywa wykonywanie pętli

Język C - pętla for (najczęstsze błędy)

- Postawienie średnika na końcu pętli **for**

```
int i;  
for (i=0; i<10; i++);  
printf("%d ", i);
```

10

- Przecinki zamiast średników pomiędzy wyrażeniami

```
int i;  
for (i=0, i<10, i++)  
    printf("%d ", i);
```

Błąd kompilacji!

Język C - pętla for (najczęstsze błędy)

- Błędny warunek - brak wykonania instrukcji

```
int i;  
for (i=0; i>10; i++)  
    printf("%d ", i);
```



- Błędny warunek - pętla nieskończona

```
int i;  
for (i=1; i>0; i++)  
    printf("%d ", i);
```

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

Język C - pętla nieskończona

```
for (wyr1; wyr2; wyr3)  
    instrukcja
```

- Wszystkie wyrażenia (**wyr1**, **wyr2**, **wyr3**) w pętli for są opcjonalne

```
for ( ; ; )  
    instrukcja
```

- pętla nieskończona

- W przypadku braku **wyr2** przyjmuje się, że jest ono **prawdziwe**

Język C - zagnieżdżanie pętli for

- Jako instrukcja w pętli **for** może występować kolejna pętla **for**

```
int i, j;
for (i=1; i<=3; i++)           // pętla zewnętrzna
    for (j=1; j<=2; j++)       // pętla wewnętrzna
        printf("i: %d    j: %d\n", i, j);
```

```
i: 1    j: 1
i: 1    j: 2
i: 2    j: 1
i: 2    j: 2
i: 3    j: 1
i: 3    j: 2
```

Język C - operator inkrementacji (++)

- Jednoargumentowy operator **++** zwiększa wartość zmiennej o 1 (nie wolno stosować go do wyrażeń)
- Operator **++** może występować jako przedrostek lub przyrostek

| Zapis | Nazwa | Znaczenie |
|------------|-------------------|---|
| ++x | preinkrementacji | wartość zmiennej jest modyfikowana przed jej użyciem |
| x++ | postinkrementacji | wartość zmiennej jest modyfikowana po użyciu jej poprzedniej wartości |

Język C - operator inkrementacji (++)

■ Przykład

```
int x = 1, y;  
y = 2 * ++x;
```

```
int x = 1, y;  
y = 2 * x++;
```

■ Kolejność operacji

```
++x          x = 2  
2 * ++x     2 * 2  
y = 2 * ++x y = 4
```

```
2 * x       2 * 1  
y = 2 * x   y = 2  
x++         x = 2
```

■ Wartości zmiennych

```
x = 2    y = 4
```

```
x = 2    y = 2
```

Język C - operator inkrementacji (++)

- Miejsce umieszczenia operatora **++** nie ma znaczenia w przypadku instrukcji typu:

```
x++;  
++x;
```

równoważne

```
x = x + 1;
```

- Nie należy stosować operatora **++** do zmiennych pojawiających się w wyrażeniu więcej niż jeden raz

```
x = x++;  
x = ++x;
```

- Zgodnie ze standardem języka C wynik powyższych instrukcji jest **niezdefiniowany**

Język C - operator dekrementacji (--)

- Jednoargumentowy operator -- zmniejsza wartość zmiennej o 1 (nie wolno stosować go do wyrażeń)
- Operator -- może występować jako przedrostek lub przyrostek

| Zapis | Nazwa | Znaczenie |
|-------|-------------------|---|
| --x | predekrementacji | wartość zmiennej jest modyfikowana przed jej użyciem |
| x-- | postdekrementacji | wartość zmiennej jest modyfikowana po użyciu jej poprzedniej wartości |

Język C - priorytet operatorów ++ i --

| Priorytet | Operator / opis |
|-----------|--|
| 1 | ++ -- (przyrostki) () [] . -> |
| 2 | ++ -- (przedrostki) sizeof (typ) + - ! ~ * & (jednoargumentowe) |
| 3 | * / % |
| 4 | + - (dwuargumentowe) |
| 5 | << >> |
| 6 | < > <= >= |
| 7 | == != |
| 8 | & (bitowy) |
| 9 | ^ |

Język C - miesięczny kalendarz

- Napisz program wyświetlający miesięczny kalendarz. Wczytaj liczbę dni w miesiącu i dzień tygodnia, od którego zaczyna się miesiąc.
- Przykład działania programu:

Liczba dni w miesiącu: 31

Pierwszy dzień tygodnia (1-Pn, 2-Wt, ...): 4

| Pn | Wt | Sr | Cz | Pt | So | N |
|----|----|----|----|----|----|----|
| | | | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 31 | |

Język C - miesięczny kalendarz

```
#include <stdio.h>

int main()
{
    int ile_dni, dzien_tyg, i;

    printf("Liczba dni w miesiacu: "); scanf("%d",&ile_dni);
    printf("Pierwszy dzien tygodnia (1-Pn, 2-Wt, ...): ");
    scanf("%d",&dzien_tyg);

    printf("-----\n");
    printf(" Pn Wt Sr Cz Pt So  N\n");

    for (i=1; i<dzien_tyg; i++) printf("  ");
    for (i=0; i<ile_dni; i++)
    {
        printf("%3d",i+1);
        if ((i+dzien_tyg)%7==0) printf("\n");
    }
    printf("\n"); return 0;
}
```

Język C - miesięczny kalendarz

```
#include <stdio.h>
int main()
{
    int ile_dni;
    int dzien_tyg;

    printf("Liczba dni w miesiącu: ");
    printf("Pierwszy dzień tygodnia (1-Pn, 2-Wt, ...): ");
    scanf("%d %d", &ile_dni, &dzien_tyg);

    printf("-----\n");
    printf("Pn Wt Sr Cz Pt So N\n");

    for (i=1; i<dzien_tyg; i++) printf("    ");
    for (i=0; i<ile_dni; i++)
    {
        printf("%3d", i+1);
        if ((i+dzien_tyg)%7==0) printf("\n");
    }
    printf("\n"); return 0;
}
```

```
Liczba dni w miesiącu: 30
Pierwszy dzień tygodnia (1-Pn, 2-Wt, ...): 5
-----
Pn Wt Sr Cz Pt So N
    1  2  3
  4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

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?

00010101000001110001010000010000



| | | | |
|----------|----------|----------|----------|
| 00010101 | 00000111 | 00010100 | 00010000 |
|----------|----------|----------|----------|

Kod binarny?



| | | | | | | | |
|------|------|------|------|------|------|------|------|
| 0001 | 0101 | 0000 | 0111 | 0001 | 0100 | 0001 | 0000 |
|------|------|------|------|------|------|------|------|

A może BCD?



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 0 | 7 | 1 | 4 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Liczba: 15 071 410 ?



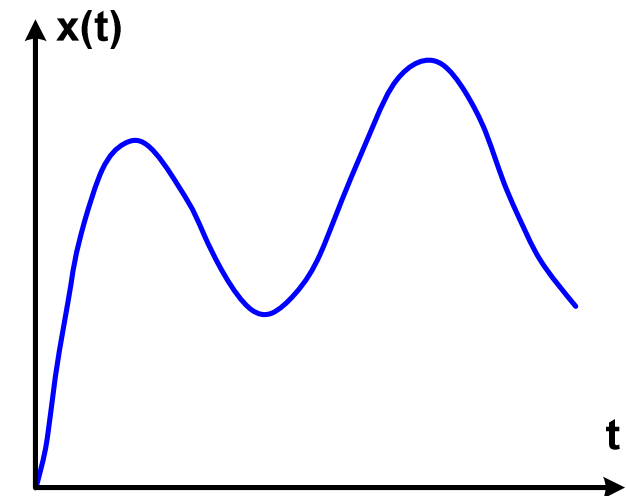
15 lipca 1410 roku

Data !!!

Informacja analogowa i cyfrowa

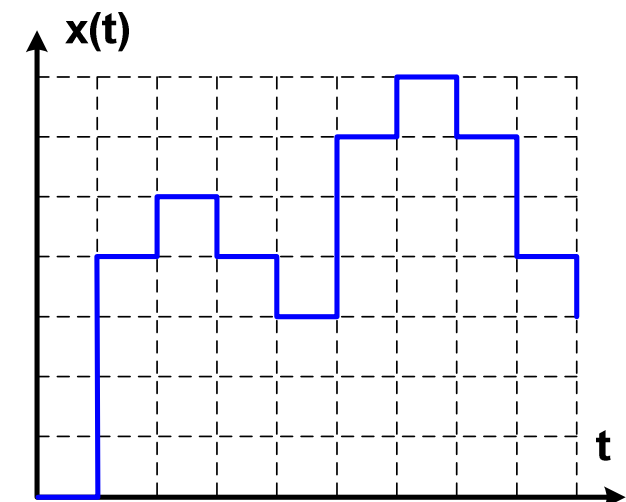
■ Sygnał analogowy

- 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 |
| <i>1</i> | <i>5</i> | <i>10</i> | <i>50</i> | <i>100</i> | <i>500</i> | <i>1000</i> |

Liczby i cyfry

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

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

- indyjsko-arabskie

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| १ | २ | ३ | ४ | ० | ६ | ७ | ८ | ९ | . |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |

- wschodnio-indyjsko-arabskie

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| १ | २ | ३ | ४ | ५ | ६ | ७ | ८ | ९ | . |
| 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

| | | | | | | | | | |
|---|---|----|----|-----|----|-----|-----|------|---|
| ┆ | ∧ | X | XX | ∧XX | ↑ | * | (C) | ⊙ | ⊙ |
| 1 | 5 | 10 | 20 | 25 | 50 | 100 | | 1000 | |

cyfry grecko-jońskie

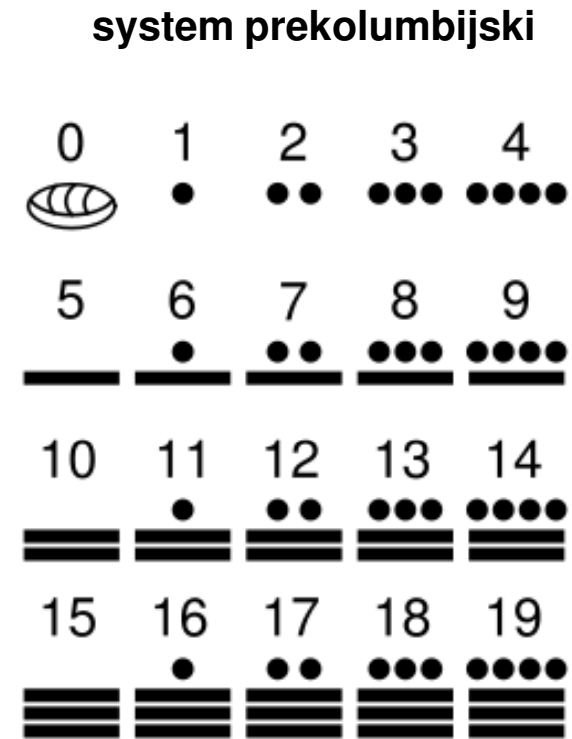
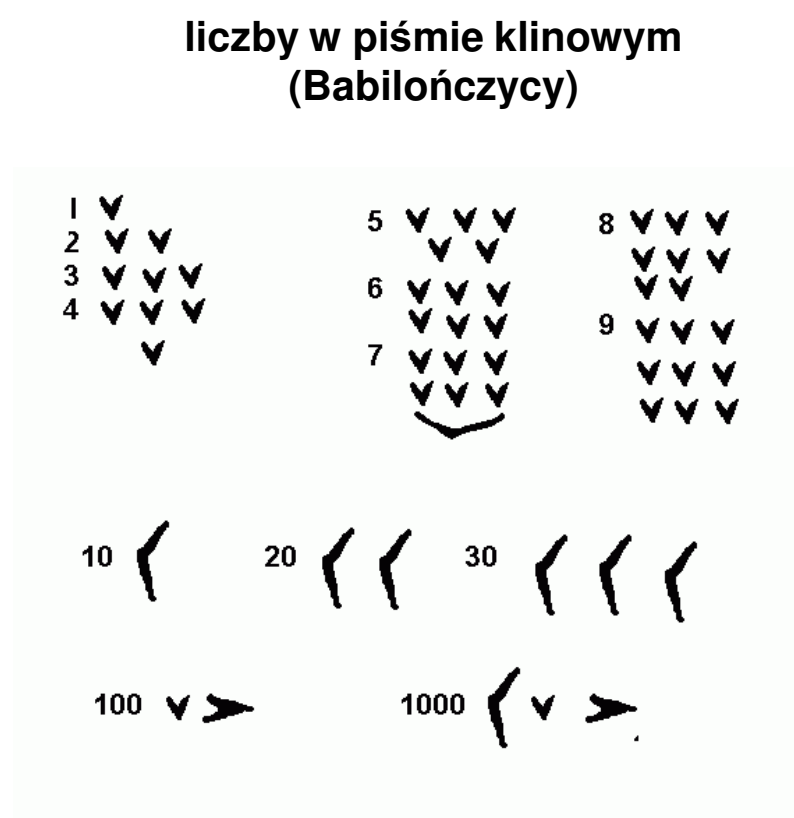
| | | | | | | | | |
|---|------|------|------|------|------|------|------|------|
| α | β | γ | δ | ε | ς | ζ | η | θ |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| ι | κ | λ | μ | ν | ξ | ο | π | ρ |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| σ | ϖ | τ | υ | φ | χ | ψ | ω | Ͱ |
| 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 |
| α' | β' | γ' | δ' | ε' | ς' | ζ' | η' | θ' |
| 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 | 8000 | 9000 |
| α'Ͱε'β' | | | | | | | | |

cyfry w pisowni chińskiej

| | | | |
|--------|---|----------|---|
| jeden | 一 | sześć | 六 |
| dwa | 二 | siedem | 七 |
| trzy | 三 | osiem | 八 |
| cztery | 四 | dziewięć | 九 |
| pięć | 五 | dziesięć | 十 |
| zero | 另 | | |

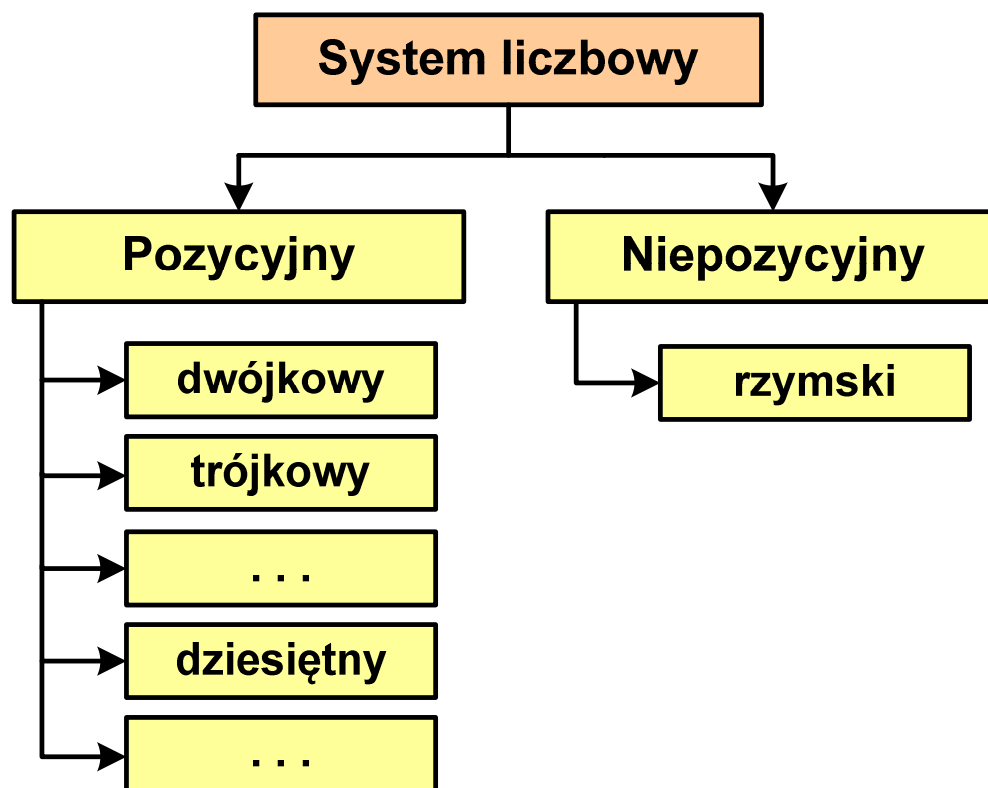
Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:



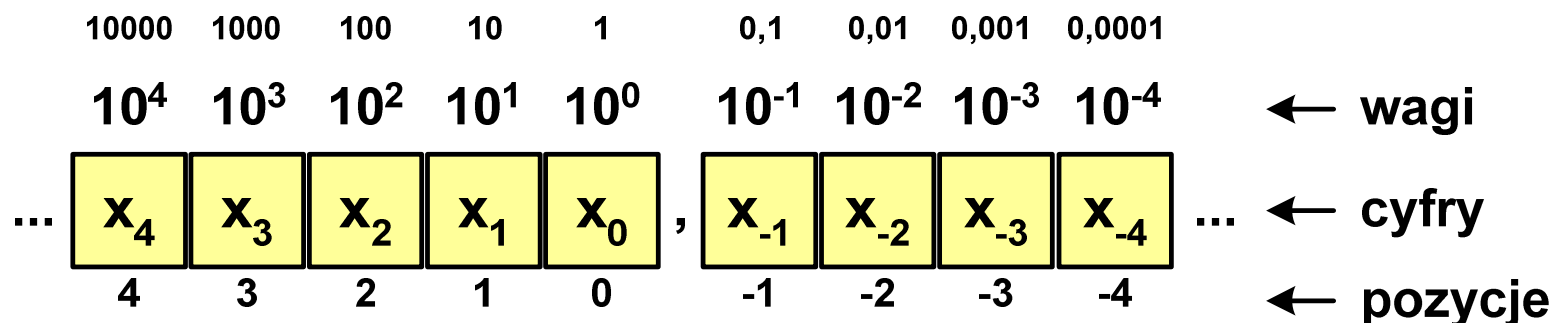
Systemy liczbowe

- **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
- $p = 10$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

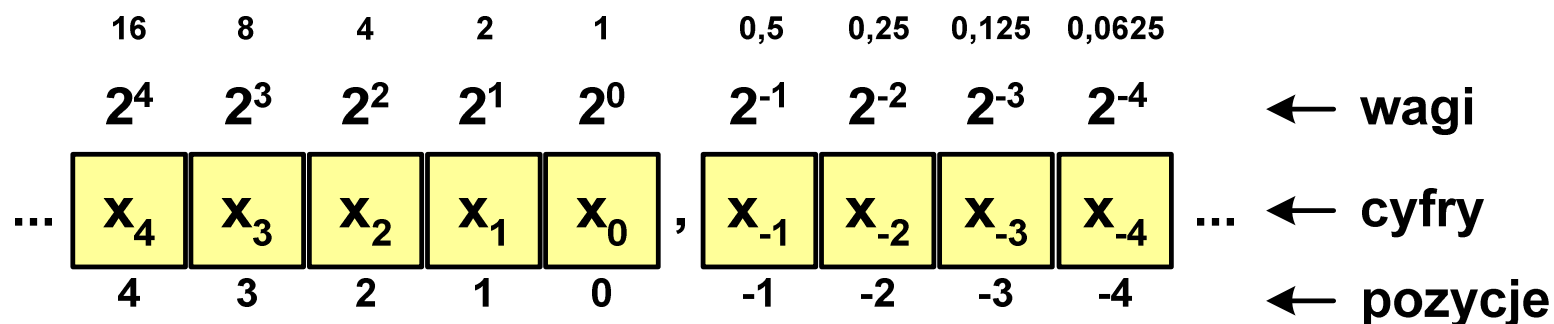
| | | | | | | | |
|--|--------|--------|--------|--------|-----------|-----------|---|
| | 10^3 | 10^2 | 10^1 | 10^0 | 10^{-1} | 10^{-2} | |
| | 1 | 4 | 0 | 8 | , | 2 | 5 |

1408,25₍₁₀₎ =

$$= 1 \cdot 10^3 + 4 \cdot 10^2 + 0 \cdot 10^1 + 8 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

$$= 1000 + 400 + 0 + 8 + 0,2 + 0,05$$

System dwójkowy (ang. binary)



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

| | | | | | | | |
|-------|-------|-------|-------|---|----------|----------|----------|
| 2^3 | 2^2 | 2^1 | 2^0 | , | 2^{-1} | 2^{-2} | 2^{-3} |
| 1 | 1 | 0 | 1 | | 1 | 0 | 1 |

1101,101₍₂₎ =

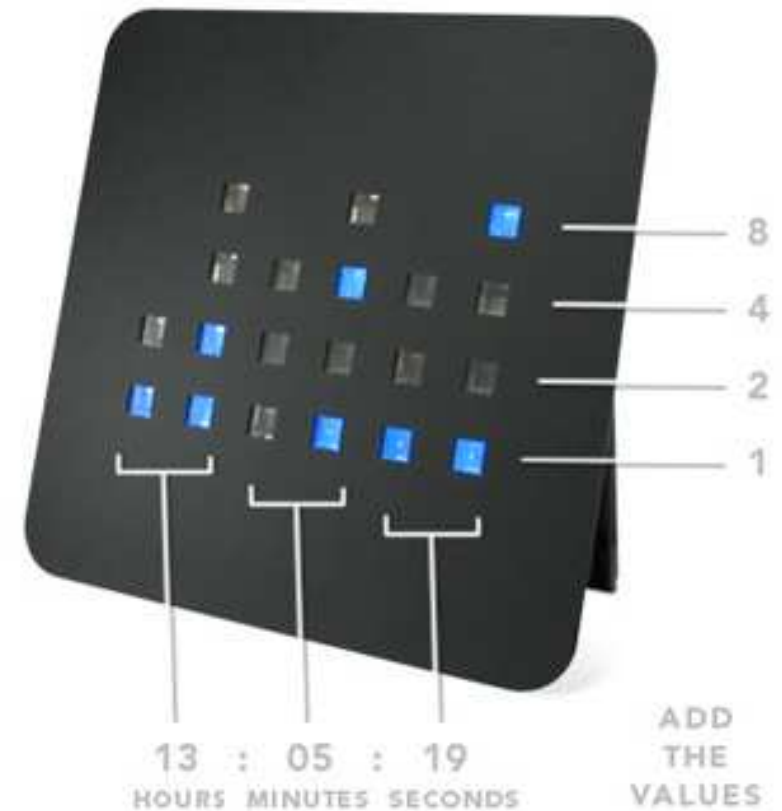
$$= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$= 8 + 4 + 0 + 1 + 0,5 + 0 + 0,125$$

$$= 13,625_{(10)}$$

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

$3A5D_{(16)}$

$3A5D_{16}$

$3A5D_{\text{hex}}$

$(3A5D)_{\text{hex}}$

$(3A5D)_{16}$

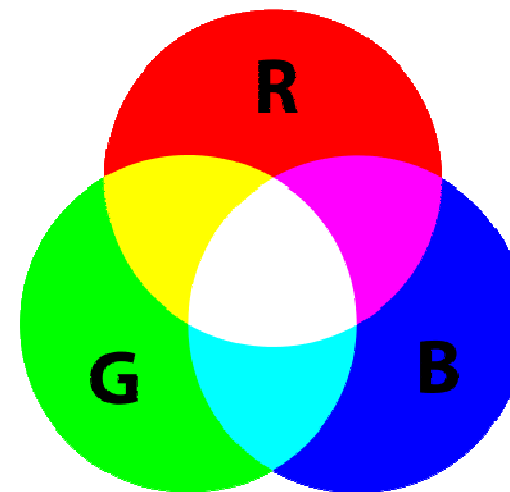
\$3A5D

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:

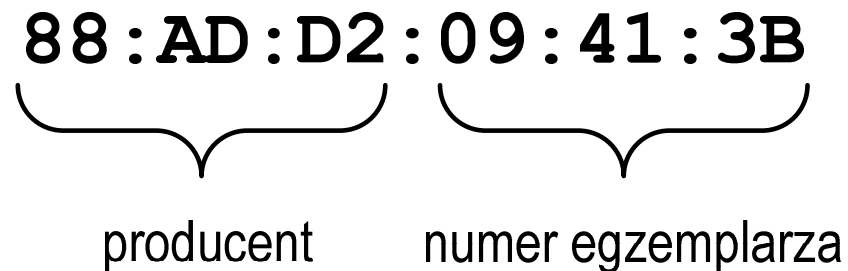
```
<BODY bgcolor="#336699" text="#000000" link="#FFFF00"  
vlink="#33FFFF" alink="#FF0000">
```

The screenshot shows a website interface with a dark blue background. On the left, there are two buttons: 'ARCHIWUM' and 'ENGLISH'. The main content area is titled 'Studia stacjonarne:' and lists the following schedule:

| | |
|----------------------|--|
| Poniedziałek: | |
| 12:15 - 14:00 | Informatyka 1 - wykład , sem. 2 ED, WE-Aula II |
| Wtorek: | |
| 08:30 - 10:00 | Informatyka 1 - prac. , sem. 2 ED, gr. PS3, WE-110 |
| 12:15 - 13:45 | Informatyka 1 - prac. , sem. 2 ED, gr. PS2, WE-110 |
| 14:15 - 15:45 | Informatyka 1 - prac. , sem. 2 ED, gr. PS1, WE-110 |

System szesnastkowy - zastosowania

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



- <http://hwaddress.com>

| OUI | MAC range | Company |
|----------|---------------------------------------|-----------------------------|
| 88-AD-D2 | 88-AD-D2-00-00-00 - 88-AD-D2-FF-FF-FF | Samsung Electronics Co.,Ltd |

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)

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

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

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

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

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

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

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

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

Konwersja z systemu dziesiętnego na dowolny

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

$$626_{(10)} = ?_{(2)}$$

$$626_{(10)} = 1001110010_{(2)}$$

| | | |
|---------------|---------------|---|
| $626/2 = 313$ | <i>reszta</i> | 0 |
| $313/2 = 156$ | <i>reszta</i> | 1 |
| $156/2 = 78$ | <i>reszta</i> | 0 |
| $78/2 = 39$ | <i>reszta</i> | 0 |
| $39/2 = 19$ | <i>reszta</i> | 1 |
| $19/2 = 9$ | <i>reszta</i> | 1 |
| $9/2 = 4$ | <i>reszta</i> | 1 |
| $4/2 = 2$ | <i>reszta</i> | 0 |
| $2/2 = 1$ | <i>reszta</i> | 0 |
| $1/2 = 0$ | <i>reszta</i> | 1 |

kolejność odczytywania
cyfr liczby w systemie
dwójkowym

kończymy, gdy liczba dziesiętna ma wartość 0

Konwersja z systemu dziesiętnego na dowolny

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

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

| | | | |
|--------------|---------------|---|---|
| $626/7 = 89$ | <i>reszta</i> | 3 | ↑ |
| $89/7 = 12$ | <i>reszta</i> | 5 | |
| $12/7 = 1$ | <i>reszta</i> | 5 | |
| $1/7 = 0$ | <i>reszta</i> | 1 | |

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

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

| | | | | |
|---------------|---------------|----|-----|---|
| $626/14 = 44$ | <i>reszta</i> | 10 | → A | ↑ |
| $44/14 = 3$ | <i>reszta</i> | 2 | | |
| $3/14 = 0$ | <i>reszta</i> | 3 | | |

Szybkie konwersje: $2 \rightarrow 4, 8, 16$ $4, 8, 16 \rightarrow 2$

$2 \rightarrow 4$

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

$$\underbrace{01}_1 \mid \underbrace{10}_2 \mid \underbrace{11}_3 \mid \underbrace{00}_0 \mid \underbrace{11}_3$$

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

$2 \rightarrow 8$

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

$$\underbrace{010}_2 \mid \underbrace{110}_6 \mid \underbrace{011}_3$$

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

$2 \rightarrow 16$

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

$$\underbrace{0101}_5 \mid \underbrace{1010}_A$$

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

$4 \rightarrow 2$

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

$$\underbrace{01}_1 \mid \underbrace{10}_2 \mid \underbrace{11}_3 \mid \underbrace{00}_0 \mid \underbrace{11}_3$$

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

$8 \rightarrow 2$

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

$$\underbrace{010}_2 \mid \underbrace{110}_6 \mid \underbrace{011}_3$$

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

$16 \rightarrow 2$

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

$$\underbrace{0101}_5 \mid \underbrace{1010}_A$$

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

Koniec wykładu nr 3

Dziękuję za uwagę!