

Architektura komputerów i systemy operacyjne

Lista 2

- 1 Zamienić liczbę dziesiętną 250, 5 na liczbę o podstawie: 3; 4; 7; 8; 16.
- 2 Zamienić liczby dziesiętne na binarne: 12, 0625; 10^4 ; 673, 23 i 2014.
- 3 Zamienić następujące liczby na liczby o innej podanej podstawie
 - a) dziesiętną 225, 225 na binarną, ósemkową i szesnastkową
 - b) binarną 11010111, 110 na dziesiętną, ósemkową i szesnastkową
 - c) ósemkową 623, 77 na dziesiętną, binarną i szesnastkową
 - d) szesnastkową $2ACD, D$ na dziesiętną, ósemkową i binarną
- Uwaga:** Zauważ, że łatwo można zamieniać liczby między podstawami 2, 2^2 , 2^3 , 2^4 . Jak?
- 4 Dla poniższych przypadków znajdź podstawę $p \in \mathbb{Z}$ lub pokaż, że taka podstawa nie istnieje:
 - a) $(BEE)_p = (2699)_{10}$
 - b) $(365)_p = (194)_{10}$
 - c) $((34)_p + (24)_p) \cdot (21)_p = (1480)_p$
- 5 Wytłumacz kodowanie BCD (ang. Binary-Coded Decimal). Przedstaw liczbę 127, 1616, 2017 w kodzie BCD.
- 6 Wytłumacz kodowanie UTF-8. Przedstaw ciąg znaków 'część' w kodzie UTF-8 w postaci szesnastkowej. Wytłumacz przekształcenie.
- 7 Wytłumacz kodowanie Greya (ang. Gray code). Pokaż konstrukcję kodów Graya 2, 3, 4 bitowych.
- 8 Skompiluj poniższy program na dowolnym kompilatorze języka C np. gcc. Wytłumacz **dokładnie** zachowanie tego programu.

```
#include <stdio.h>

int main(int argc, char **argv)
{
    if ( (1.1 - 1.0)*10.0 - 1.0 == 0.0 )
        printf("Prawda\n");
    else
        printf("Fałsz\n");

    return 0;
}
```

9 (*) Wy tłumacz co robi poniższy program (przeprowadzając dokładną analizę)

```
float InvSqrt(float x)
{
    float xhalf = 0.5f*x;
    int i = *(int*)&x;
    i = 0x5f3759df - (i>>1);
    x = *(float*)&i;
    x = x*(1.5f-xhalf*x*x);
    return x;
}
```

Wskazówka:

- <https://web.archive.org/web/20220805032430/https://cs.uwaterloo.ca/~m32rober/rsqrt.pdf>
- http://en.wikipedia.org/wiki/Fast_inverse_square_root