

DOKUMENTACE PROGRAMU

Převod čísla $c, c \in \mathbb{Z}^+$, z binární soustavy do hexadecimální a naopak

Autor: Jan Šindelář

Předmět: Úvod do programování

Ročník: 2. B-FGG

Datum odevzdání: 12.2.2026

Obsah

1. Zadání
2. Úvod
3. Specifikace úlohy
4. Teoretický základ
5. Návrh řešení
6. Popis implementace
7. Uživatelská dokumentace
8. Omezení a možná rozšíření
9. Závěr
10. Seznam použité literatury

1. Zadání

Zadané kladné reálné číslo převeďte z dvojkové (binární) soustavy do soustavy šestnáctkové (hexadecimální) a zpět. Nepoužívejte konverzní funkce v jazyce Python.

2. Úvod

Tento dokument popisuje návrh, tvorbu a použití programu v jazyce Python, který slouží k převodu čísel mezi binární (dvojkovou) a hexadecimální (šestnáctkovou) číselnou soustavou. Dokumentace je určena jak pro uživatele programu, tak pro vývojáře, kteří by jej chtěli dále rozšiřovat.

Program je realizován jako konzolová aplikace. Uživatel si zvolí směr převodu a zadá číslo v příslušné soustavě. Program provede kontrolu vstupu, zajistí správný převod a vypíše výsledek.

Cílem programu je demonstrovat: práci s funkcemi v Pythonu, validaci vstupních dat, práci s řetězci, použití slovníků pro mapování hodnot a základní algoritmické postupy při převodech číselných soustav.

3. Specifikace úlohy

3.1 Zadání

Vytvořit program, který:

- umožní uživateli vybrat směr převodu,
- převede číslo z binární soustavy do hexadecimální,
- převede číslo z hexadecimální soustavy do binární,
- ověří platnost vstupu,
- ošetří chybné zadání.

3.2 Vstupy programu

Program vyžaduje následující vstupy:

1. Volba směru převodu

- 1 – převod z binární do hexadecimální soustavy
- 2 – převod z hexadecimální do binární soustavy

2. Číselná hodnota

- řetězec reprezentující číslo v odpovídající soustavě

Příklad vstupu:

What will we convert? 1 = from binary to hexadecimal, 2 = from hexadecimal to binary: 1

Enter a number in binary system: 101011

3.3 Výstupy programu

Program vypíše:

- převedenou hodnotu v cílové soustavě,
- chybovou zprávu v případě neplatného vstupu.

Příklad výstupu:

Hexadecimal form: 2B

2.4 Chybové stavy

Program ošetřuje:

- zadání jiného znaku než 0 a 1 u binárního čísla,
- zadání jiného znaku než 0–9, A–F u hexadecimálního čísla,
- špatně zvolenou možnost převodu.

4. Teoretický základ

4.1 Binární soustava

Binární soustava je poziční číselná soustava o základu 2. Používá pouze dvě číslice:

0, 1

Každá pozice čísla představuje mocninu čísla 2.

Například:

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

4.2 Hexadecimální soustava

Hexadecimální soustava je poziční soustava o základu 16. Používá číslice:

0–9 a A–F

kde:

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

4.3 Vztah mezi binární a hexadecimální soustavou

Jedna hexadecimální číslice odpovídá přesně čtyřem binárním bitům:

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

Tato přímá vazba umožňuje jednoduchý převod pomocí mapování čtveřic bitů na hexadecimální číslice a naopak.

5. Návrh řešení

5.1 Struktura programu

Program je rozdělen do tří částí:

1. Funkce bin_to_hex(n)
2. Funkce hex_to_bin(n)
3. Hlavní řídicí část programu

Tento přístup zvyšuje přehlednost a modularitu kódu.

5.2 Funkce bin_to_hex

Postup:

1. Odstranění nadbytečných mezer.
2. Kontrola platnosti vstupu.
3. Doplnění nul zleva tak, aby délka byla násobkem 4.
4. Rozdělení na čtveřice.
5. Převod pomocí mapovacího slovníku.
6. Odstranění počátečních nul.

5.3 Funkce hex_to_bin

Postup:

1. Odstranění mezer a převod na velká písmena.
2. Kontrola platnosti znaků.
3. Mapování každé hexadecimální číslice na čtyřbitovou binární reprezentaci.
4. Odstranění počátečních nul.

5.4 Datové struktury

Použit je slovník (dictionary) pro mapování:

- binární čtveřice → hexadecimální číslice
- hexadecimální číslice → binární čtveřice

Tato struktura umožňuje rychlý a přehledný převod bez nutnosti výpočtu přes desítkovou soustavu.

6. Popis implementace

6.1 Zdrojový kód

```
# Binary to hexadecimal conversion function
```

```
def bin_to_hex(n):
```

```
    n = n.strip()
```

```
# Input validity check
```

```
for sign in n:
```

```
    if sign not in "01":
```

```
        print("Error: Invalid binary number.")
```

```
        exit()
```

```
# Padding zeros from the left to make the length a multiple of 4
```

```
while len(n) % 4 != 0:
    n = "0" + n

# Binary quads to hexadecimal digit conversion map
map = {
    "0000": "0", "0001": "1", "0010": "2", "0011": "3",
    "0100": "4", "0101": "5", "0110": "6", "0111": "7",
    "1000": "8", "1001": "9", "1010": "A", "1011": "B",
    "1100": "C", "1101": "D", "1110": "E", "1111": "F"
}

hexadecimal = ""

# Binary to hexadecimal conversion
for i in range(0, len(n), 4):
    quad = n[i:i+4]
    hexadecimal += map[quad]

# Removing leading zeros from a number
while len(hexadecimal) > 1 and hexadecimal[0] == "0":
    hexadecimal = hexadecimal[1:]

return hexadecimal

# Hexadecimal to binary conversion function
def hex_to_bin(n):
    n = n.strip().upper()

    # Input validity check
    valid = "0123456789ABCDEF"
    for sign in n:
```

```

if sign not in valid:
    print("Error: Invalid hexadecimal number.")
    exit()

# Hexadecimal digit to binary quads conversion map
map = {
    "0": "0000", "1": "0001", "2": "0010", "3": "0011",
    "4": "0100", "5": "0101", "6": "0110", "7": "0111",
    "8": "1000", "9": "1001", "A": "1010", "B": "1011",
    "C": "1100", "D": "1101", "E": "1110", "F": "1111"
}

binar = ""

# Hexadecimal to binary conversion
for sign in n:
    binar += map[sign]

# Removing leading zeros from a number
while len(binar) > 1 and binar[0] == "0":
    binar = binar[1:]

return binar

# Entering input from the user and outputting the result
vyber = int(input("What will we convert? 1 = from binary to hexadecimal, 2 = from hexadecimal to binary:"))

if vyber == 1:
    bin_cislo = input("Enter a number in binary system: ")
    print("Hexadecimal form:", bin_to_hex(bin_cislo))

```

```
elif vyber == 2:  
    hex_cislo = input("Enter a number in hexadecimal system: ")  
    print("Binary form:", hex_to_bin(hex_cislo))  
  
else:  
    print("Incorrect selection parameters entered. Please select option 1 or 2..")
```

6.2 Ošetření vstupu

Platnost vstupu je kontrolována průchodem každého znaku řetězce. V případě chyby je vypsána chybová hláška ("Error: Invalid binary number.") nebo ("Error: Invalid hexadecimal number.") a program je ukončen pomocí exit().

5.3 Řídicí část programu

Program využívá podmínku if-elif-else pro výběr správné větve podle uživatelovy volby. Pokud uživatel zadá nesprávné číslo volby, je vypsána chybová hláška ("Incorrect selection parameters entered. Please select option 1 or 2..").

7. Uživatelská dokumentace

7.1 Požadavky na prostředí

- Python 3.x
- Konzolové prostředí

7.2 Spuštění programu

Program se spouští z příkazové řádky: python prevod_cisel.py

7.3 Postup použití

1. Zvolte možnost převodu (1 nebo 2).
2. Zadejte číslo v odpovídající soustavě.
3. Program vypíše převedenou hodnotu.

8. Omezení a možná rozšíření

7.1 Omezení

- Program nepodporuje záporná čísla.
- Nepodporuje desetinné zlomky.
- Nezpracovává velmi velká čísla optimalizovaným způsobem.
- Při chybě používá exit(), což není ideální řešení pro větší aplikace.

8.2 Možná rozšíření

- Podpora převodu přes desítkovou soustavu.
- Podpora dalších číselných soustav.
- Ošetření výjimek místo použití exit().
- Grafické uživatelské rozhraní.
- Jednotkové testy.

9. Závěr

Program splňuje zadání a umožňuje obousměrný převod mezi binární a hexadecimální soustavou. Využívá jednoduchý, ale efektivní princip mapování čtveřic bitů na hexadecimální číslice. Implementace je přehledná, modulární a vhodná pro výukové účely.

Řešení demonstруje práci s funkcemi, slovníky a validací vstupu v jazyce Python.

10. Seznam použité literatury

KRYL, Tomáš. *Dokumentace programů* [online]. KSVI MFF UK. Dostupné z: <https://ksvi.mff.cuni.cz/~kryl/dokumentace.htm> [cit. 12. 2. 2026].

PYTHON SOFTWARE FOUNDATION. *Python 3 Documentation* [online]. Python documentation verze 3.14.3. © 2001–2026 Python Software Foundation. Dostupné z: <https://docs.python.org/3/> [cit. 12. 2. 2026].

LUTZ, Mark. *Learning Python*. 5th ed. Sebastopol: O'Reilly Media, 2013. ISBN 978-1-4493-5573-9. [online]. Dostupné z: https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf [cit. 12. 2. 2026].