AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES

(AIMS RWANDA, KIGALI)

Group : 6	Project : 6
Course: Python Programming	Date: October 8, 2025

Hexadecimal and Decimal Number System Converter

Group Members:

Belyse KANEZA
Foumban Marah Khadij NOURIA
Philomene CYUBAHIRO
Joachim SINYABE DANBE

Contents

Abstract

This report presents our implementation of a bidirectional converter between hexadecimal and decimal number systems using pure Python. Our program provides an interactive tool for converting numbers between these two important number systems, with robust error handling and user-friendly interface. The project demonstrates practical programming skills including algorithm implementation, exception handling, and user input validation.

1 Introduction

1.1 Project Overview

We developed a Python program that converts between hexadecimal and decimal number systems through two main functions:

- `hex2int`: Converts hexadecimal strings to decimal integers
- `int2hex`: Converts decimal integers to hexadecimal strings

The challenge was to implement these conversions manually without using Python's built-in functions like `int(x, 16)` or `hex()`.

1.2 Why This Matters

Hexadecimal numbers are widely used in:

- Computer programming and memory addressing
- Digital electronics and color codes (RGB)
- Network protocols and system debugging
- File formats and data encoding

Understanding how to convert between number systems is fundamental in computer science.

2 Theoretical Background

2.1 Number Systems

2.1.1 Decimal System (Base-10)

The decimal system uses 10 digits (0-9) and is a positional notation system where each position represents a power of 10:

$$\text{Number}_{10} = d_n \times 10^n + d_{n-1} \times 10^{n-1} + \cdots + d_1 \times 10^1 + d_0 \times 10^0$$

2.1.2 Hexadecimal System (Base-16)

The hexadecimal system uses 16 symbols (0-9, A-F) where each position represents a power of 16:

$$\text{Number}_{16} = h_n \times 16^n + h_{n-1} \times 16^{n-1} + \cdots + h_1 \times 16^1 + h_0 \times 16^0$$

Where: A=10, B=11, C=12, D=13, E=14, F=15

2.2 Conversion Algorithms

2.2.1 Hexadecimal to Decimal Conversion

The algorithm processes each hexadecimal digit from left to right:

$$\text{Decimal} = \sum_{i=0}^{n-1} (\text{digitValue}_i \times 16^{(n-1-i)})$$

Where:

- n is the number of digits
- digitValue_i is the decimal value of the hexadecimal digit at position i
- Positions are numbered from left (most significant) to right (least significant)

2.2.2 Decimal to Hexadecimal Conversion

The algorithm uses repeated division:

1. Divide the decimal number by 16
2. The remainder becomes the least significant hexadecimal digit
3. The quotient becomes the new number for the next iteration
4. Repeat until quotient becomes zero
5. The hexadecimal digits are collected in reverse order

3 Our Implementation

3.1 How It Works

Our program follows a simple but effective design:

1. User chooses conversion direction through a menu
2. Input is validated for correctness
3. Conversion algorithm processes the input
4. Result is displayed with proper formatting
5. Errors are caught and handled gracefully

3.2 Key Features

- **Bidirectional Conversion**: Both hex-to-decimal and decimal-to-hex
- **Error Handling**: Catches invalid inputs with clear messages
- **User-Friendly**: Simple menu interface
- **Case Insensitive**: Accepts both uppercase and lowercase hex digits
- **Multi-digit Support**: Handles numbers of any length

3.3 Core Functions

3.3.1 Hexadecimal to Decimal Conversion

Listing 1: hex2int Function

```
def hex2int(hex_str):
    """
    This function convert a hexadecimal string to a decimal integer.

    input :
    hex_string (str): A string containing one or more hexadecimal digits (0-9, A-F, a-f).

    Output:
    int: The corresponding decimal integer value.

    Raises:
    ValueError: If the input string contains invalid hexadecimal digits.
    """
    hex_str = hex_str.upper()
    hex_digits = "0123456789ABCDEF"

    result = 0

    for i, char in enumerate(hex_str):
        if char not in hex_digits:
            raise ValueError(f"'{char}' is not a valid hex digit")

        digit_value = hex_digits.index(char)
        power = len(hex_str) - 1 - i
        result += digit_value * (16 ** power)

    return result
```

What it does:

- Converts input to uppercase for consistency
- Checks each character is a valid hex digit (0-9, A-F)

- Calculates the decimal value using positional weights
- Raises clear error messages for invalid inputs

3.3.2 Decimal to Hexadecimal Conversion

Listing 2: int2hex Function

```python
def int2hex(number):
    """
    This function convert a non-negative integer to its hexadecimal string representation (uppercase).

    input :
    num (int): An integer greater or equal to 0.

    Output:
    str: The hexadecimal string representing the integer.

    Raises:
    TypeError: If the input is not an integer.
    """

    if not number.isdigit():
        raise TypeError("Input must be an integer")

    number = int(number)
    if number == 0:
        return "0"

    hex_digits = "0123456789ABCDEF"
    result = ""

    n = number
    while n > 0:
        remainder = n % 16
        hex_digit = hex_digits[remainder]
        result = hex_digit + result
        n = n // 16

    return result
```

What it does:

- Validates input is a positive integer
- Handles special case of zero
- Uses repeated division by 16 to build the hex string
- Builds the result from right to left (least to most significant digit)

3.4 Error Handling

We implemented robust error handling using Python’s exception mechanism:

Listing 3: Error Handling Example

```python
try:
    result = hex2int(hex_input)
    print(f"Hexadecimal '{hex_input}' = Decimal {result}")
except ValueError as e:
    print(f"Error: {e}")
```

Errors we handle:

- Invalid hexadecimal characters (like ’G’, ’Z’)
- Non-integer inputs for decimal conversion
- Invalid menu choices

4 Testing Our Program

4.1 Sample Conversions

Here are some examples of our program in action:

Input	Output
Hexadecimal: ”A”	Decimal: 10
Hexadecimal: ”1F”	Decimal: 31
Hexadecimal: ”FF”	Decimal: 255
Decimal: 26	Hexadecimal: ”1A”
Decimal: 255	Hexadecimal: ”FF”
Decimal: 1000	Hexadecimal: ”3E8”

4.2 User Interface

Our program features a simple, intuitive menu system:

Listing 4: Interactive Menu

```
Hexadecimal and Decimal Converter
========================================

Choose an option:
1. Hexadecimal to Decimal
2. Decimal to Hexadecimal
3. Exit

Enter your choice (1-3): 1
Enter hexadecimal number: 1A
Hexadecimal '1A' = Decimal 26
```

5 What We Learned

This project helped us understand:

- How number systems work fundamentally
- Implementing algorithms from scratch
- Proper error handling with try-except blocks
- Creating user-friendly interfaces
- Testing and validating program functionality

6 Conclusion

We successfully created a functional and reliable number system converter that:

- Converts accurately between hexadecimal and decimal
- Handles errors gracefully with clear messages
- Provides an easy-to-use interface
- Demonstrates good programming practices

This project gave us valuable hands-on experience with algorithm implementation, exception handling, and creating practical software tools.

Complete Source Code

Listing 5: Our Complete Program

```python
def hex2int(hex_str):
    """
    This function convert a hexadecimal string to a decimal integer.

    input :
    hex_string (str): A string containing one or more hexadecimal digits (0-9, A-F, a-f).

    Output:
    int: The corresponding decimal integer value.

    Raises:
    ValueError: If the input string contains invalid hexadecimal digits.
    """
    hex_str = hex_str.upper()
    hex_digits = "0123456789ABCDEF"

    result = 0

    for i, char in enumerate(hex_str):
        if char not in hex_digits:
            raise ValueError(f"'{char}' is not a valid hex digit")

        digit_value = hex_digits.index(char)
        power = len(hex_str) - 1 - i
        result += digit_value * (16 ** power)

    return result

def int2hex(number):
    """
    This function convert a non-negative integer to its hexadecimal string representation (uppercase).

    input :
    num (int): An integer greater or equal to 0.

    Output:
    str: The hexadecimal string representing the integer.

    Raises:
    TypeError: If the input is not an integer.
    """

    if not number.isdigit():
        raise TypeError("Input must be an integer")
    number = int(number)
    if number == 0:
        return "0"

    hex_digits = "0123456789ABCDEF"
    result = ""

    n = number
    while n > 0:
        remainder = n % 16
        hex_digit = hex_digits[remainder]
        result = hex_digit + result
        n = n // 16

    return result

print("Hexadecimal and Decimal Converter")
print("=" * 40)

while True:
    print("\nChoose an option:")
    print("1. Hexadecimal to Decimal")
    print("2. Decimal to Hexadecimal")
    print("3. Exit")

    choice = input("Enter your choice (1-3): ")

    if choice == ’1’:
        hex_input = input("Enter hexadecimal number: ")
        try:
            result = hex2int(hex_input)
            print(f"Hexadecimal ’{hex_input}’ = Decimal {result}")
        except ValueError as e:
            print(f"Error: {e}")
    elif choice == ’2’:
        dec_input = input("Enter decimal number: ")
        try:
            result = int2hex(dec_input)
            print(f"Decimal {dec_input} = Hexadecimal ’{result}’")
        except (ValueError, TypeError) as e:
            print(f"Error: {e}")

    elif choice == ’3’:
        print("Exiting the program!!!!!!!!!")
        break

    else:
        print("Invalid choice. Please enter 1, 2, or 3.")
```