Computing Science Assignment 2 on:

Flowchart: Converting Hexadecimal to Decimal

Department of Computer Science Section RCD/2017/D

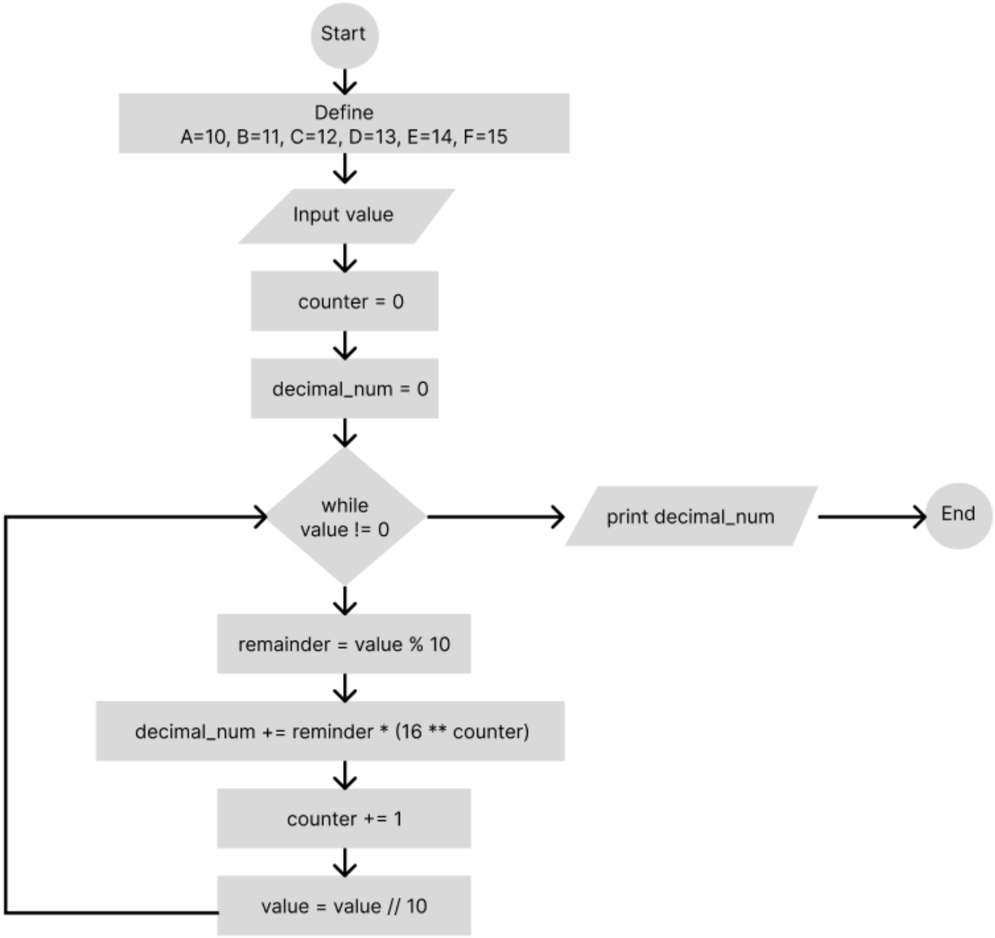
Group 1 Members

Name	ID
1. Nehimya Mesfin	RCD/0620/2017
2. Abigiya Yoseph	RCD/0590/2017
3. Dagmawit Hagos	RCD/0596/2017
4. Elshaday Abebayew	RCD/0597/2017
5. Samrawit Dawit	RCD/0625/2017
6. Saron Abenet	RCD/0626/2017
7. Sofanit Ashebir	RCD/1461/2017
8. Mohamed Zekaria	RCD/0615/2017

Submission Date: 23rd Feb, 2025

Submitted To: Mr. Dagmawi Y.

Flowchart of Hexadecimal to Decimal Conversion



Explanation of the Flowchart Step by Step:

1. **Define the values of A, B, C, D, E, F:**
   - Since the user inputs hexadecimal number the alphabetical values must be defined to maintain its flow while working on it.
2. **Enter a value:**
   - A value is input, which is assumed to represent a hexadecimal number in base 16.
3. **Initialize variables:**
   - counter = 0: This variable keeps track of the position of the digit in the hexadecimal number (starting from the least significant digit).
   - decimal_num = 0: This will store the final decimal (base 10) equivalent of the hexadecimal number.
4. **Check if value != 0:**
   - A while loop is used to process the digits of the number until the value becomes 0 (i.e., all digits are processed).
5. **Extract the last digit (remainder):**
   - remainder = value % 10: The last digit of the current value is extracted using the modulus operator (%).
6. **Update the decimal number:**
   - The decimal equivalent of the current hexadecimal digit is calculated and added to decimal_num:

     decimal_num += remainder * (16 ** counter)

     Here, remainder is multiplied by 16 raised to the power of counter, which corresponds to the positional weight of the digit in hexadecimal format.
7. **Increment the counter:**
   - counter is incremented by 1 to move to the next higher positional digit in the hexadecimal number.
8. **Remove the last digit:**
   - value = value // 10: The last digit of the value is removed using integer division (//).
9. **Repeat:**
   - Steps 4–8 are repeated until value becomes 0.
10. **Output the result:**
    - The final value of decimal_num (decimal equivalent of the hexadecimal number) is printed.

Example

Input:

Let’s take value = 25 (hexadecimal equivalent of 2 * 16^1 + 5 * 16^0).

Step-by-Step Execution:

1. **Initial Values:**
    - value = 25
    - counter = 0
    - decimal_num = 0
2. **First Iteration:**
    - remainder = value % 10 = 25 % 10 = 5
    - decimal_num += remainder * (16 ** counter) = 0 + 5 * (16 ** 0) = 5
    - counter = 0 + 1 = 1
    - value = value // 10 = 25 // 10 = 2
3. **Second Iteration:**
    - remainder = value % 10 = 2 % 10 = 2
    - decimal_num += remainder * (16 ** counter) = 5 + 2 * (16 ** 1) = 5 + 32 = 37
    - counter = 1 + 1 = 2
    - value = value // 10 = 2 // 10 = 0
4. **End Condition:**
    - Since value = 0, the loop ends.
    - Final decimal_num = 37.

Output::

The decimal equivalent of hexadecimal 25 (in numeric representation) is 37.