Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Class:	**SE**	**Semester:**	**III**
Course Code:	**2113114**	**Course Name:**	**Computer Organization & Architecture**

Name of Student:	**Kenil Anit Vanjara**
Roll No. :	72
Division:	**1**
Experiment No.:	**01**
Title of Experiment:	**Implement a program to convert hexadecimal, decimal number to binary number.**
Date of Submission:	
Date of Correction:	

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Shilpa Jaiswal

Signature :

Date :

Experiment No. 1
Implement a program to convert hexadecimal, decimal number to binary number.
Date of Performance:
Date of Correction:

Aim: To implement a program that converts numbers from hexadecimal and decimal number systems into their equivalent binary representation.

Objective: To understand and implement the conversion of decimal and hexadecimal numbers into binary format, which is fundamental to data representation in digital systems.

Theory:

In computer systems, data is always processed and stored in binary—a base-2 numeral system that uses only two digits: 0 and 1. However, humans commonly use decimal (base-10), and system-level programs or hardware diagnostics frequently rely on hexadecimal (base-16) due to its compactness and readability.

Number Systems Overview:

Number System	**Base**	**Digits Used**	**Common Usage**
Binary	2	0, 1	Used internally by all computers
Decimal	10	0–9	Used by humans for general calculations
Hexadecimal	16	0–9, A–F	Used in memory addressing, debugging, etc.

Decimal to Binary Conversion:

Decimal to binary conversion involves repeated division by 2:

- Divide the decimal number by 2.
- Store the remainder.
- Repeat the process on the quotient until the quotient is 0.
- The binary number is the remainders read in reverse.

Example:
Convert 13 to binary:

13 ÷ 2 = 6 remainder 1

6 ÷ 2 = 3 remainder 0

3 ÷ 2 = 1 remainder 1

1 ÷ 2 = 0 remainder 1

Binary = 1101

Hexadecimal to Binary Conversion:

Hexadecimal to binary conversion is direct and efficient, as each hex digit maps exactly to a 4-bit binary number.

Hex Digit	Binary Equivalent
0	0000
1	0001
...	...
A (10)	1010
F (15)	1111

Example:
Convert Hex 2F to Binary:
2 = 0010, F = 1111 → Binary = 00101111

Why Binary?

- Binary aligns with the ON/OFF (high/low voltage) nature of digital electronics.
- It simplifies the design of hardware logic circuits using gates.
- All information (text, numbers, audio, video) in digital systems is represented in binary format.

Real-World Relevance:

- Microprocessors handle data and instructions in binary format.
- Hexadecimal simplifies representation of large binary numbers in system diagnostics and debugging.
- Understanding conversions is essential for memory management, instruction decoding, and low-level programming.

Solution:

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

// Function to convert decimal number to binary
void decToBin(int num) {
    if (num == 0) {
        printf("0");
        return;
    }

    int binary[64];
    int index = 0;

    while (num > 0) {
        binary[index] = num % 2;
        num = num / 2;
        index++;
    }
```

```
    // Print binary in reverse order
    for (int j = index - 1; j >= 0; j--) {
        printf("%d", binary[j]);
    }
}

// Function to convert hexadecimal string to decimal integer
int hexToDec(char hex[]) {
    int decimal = 0, base = 1, i = 0;

    // Find length of hex string
    while (hex[i] != '\0') {
        i++;
    }
    i--;

    // Convert hex to decimal
    while (i >= 0) {
        if (hex[i] >= '0' && hex[i] <= '9') {
            decimal += (hex[i] - '0') * base;
        } else if (hex[i] >= 'A' && hex[i] <= 'F') {
```

```
            decimal += (hex[i] - 'A' + 10) * base;
        } else if (hex[i] >= 'a' && hex[i] <= 'f') {
            decimal += (hex[i] - 'a' + 10) * base;
        } else {
            printf("Invalid hexadecimal input.\n");
            return -1;
        }
        base *= 16;
        i--;
    }
    return decimal;
}

int main() {
    int choice;
    printf("=== Number System Converter ===\n");
    printf("1. Hexadecimal → Binary\n");
    printf("2. Decimal     → Binary\n");
    printf("Enter your choice (1 or 2): ");
    scanf("%d", &choice);
```

```
if (choice == 1) {
    char hex[50];
    printf("\nEnter a hexadecimal number: ");
    scanf("%s", hex);

    int decimal = hexToDec(hex);
    if (decimal != -1) {
        printf("Decimal equivalent: %d\n", decimal);
        printf("Binary equivalent : ");
        decToBin(decimal);
        printf("\n");
    }
} else if (choice == 2) {
    int dec;
    printf("\nEnter a decimal number: ");
    scanf("%d", &dec);

    printf("Binary equivalent : ");
    decToBin(dec);
    printf("\n");
} else {
```

```
        printf("\n  Invalid choice. Please select 1 or 2.\n");
    }
    return 0;
}
```

OUTPUT:

```
=== Number System Converter ===
1. Hexadecimal → Binary
2. Decimal     → Binary
Enter your choice (1 or 2): 1

Enter a hexadecimal number: ff
Decimal equivalent: 255
Binary equivalent : 11111111
```

Conclusion: We learned how to convert numbers from decimal and hexadecimal systems into binary. This helped us understand how computers represent and process different number systems internally.