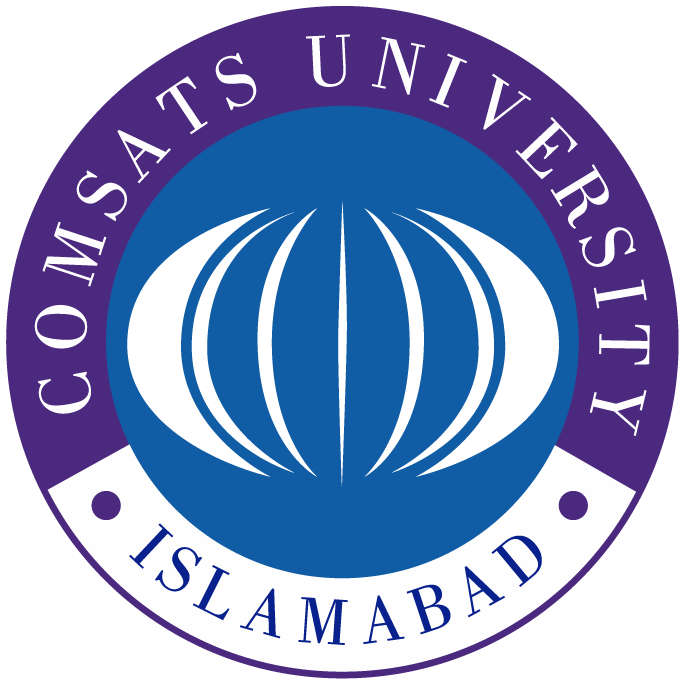
**ICT Assignment**



Submitted by: **Ali Hassan**

Reg No: **SP23-BCS-014**

Submitted to: **Dr Jawad Shafi**

**Department of Computer Science Comsats University Islamabad, Lahore Campus.**

**Problem :** Write a C programme that converts Binary , Octal , and Hexadecimal number to Decimal number.

**Source Code**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int **main**()

{

    char hex[17];

    int dec = 0, val, len;

    int num, base, rem, i = 0, tempnum;

**printf**("Enter your base : ");

**scanf**("%d", &base);

    switch (base)

    {

    case 2:

**printf**("Enter a binary number: ");

**scanf**("%d", &*num*);

*// check if entered binary number is valid*

        tempnum = num;

        while (tempnum != 0)

        {

            rem = tempnum % 10;

            if (rem != 0 && rem != 1)

            {

**printf**("Error: Entered number is not a binary number.\nEnter your number again : ");

**scanf**("%d", &num);

                tempnum = num;

            }

            tempnum /= 10;

        }

*// perform binary to decimal conversion*

        tempnum = num;

        while (tempnum != 0)

        {

            rem = tempnum % 10;

            dec += rem \* **pow**(base, i);

            i++;

            tempnum /= 10;

        }

**printf**("Decimal = %d\n", dec);

        break;

    case 8:

**printf**("Enter your number : ");

**scanf**("%d", &*num*);

*// Check the validity of octal number*

        tempnum = num;

        while (tempnum != 0)

        {

            rem = tempnum % 10;

            if (rem >= 8)

            {

**printf**("Error: Entered number is not a binary number.\nEnter your number again : ");

**scanf**("%d", &num);

                tempnum = num;

            }

            tempnum /= 10;

        }

*// perform octal to decimal conversion*

        tempnum = num;

        while (tempnum != 0)

        {

            rem = tempnum % 10;

            dec += rem \* **pow**(base, i);

            i++;

            tempnum /= 10;

        }

**printf**("Decimal = %d\n", dec);

        break;

    case 16:

**printf**("Enter a hexadecimal number : ");

**scanf**("%s", &*hex*);

        len = **strlen**(hex);

        len--;

        for (i = 0; hex[i] != 0; i++)

        {

            if (hex[i] < '0' || hex[i] > '9' && hex[i] < 'A' || hex[i] > 'F')

            {

**printf**("Error.Invalid Number.");

                dec = 0;

                break;

            }

            if (hex[i] >= '0' && hex[i] <= '9')

            {

                val = hex[i] - 48;

            }

            else if (hex[i] >= 'A' && hex[i] <= 'F')

            {

                val = hex[i] - 65 + 10;

            }

            dec = dec + val \* **pow**(16, len);

            len--;

        }

        if (dec != 0)

        {

**printf**("Decimal value = %d", dec);

            break;

        }

        break;

    default:

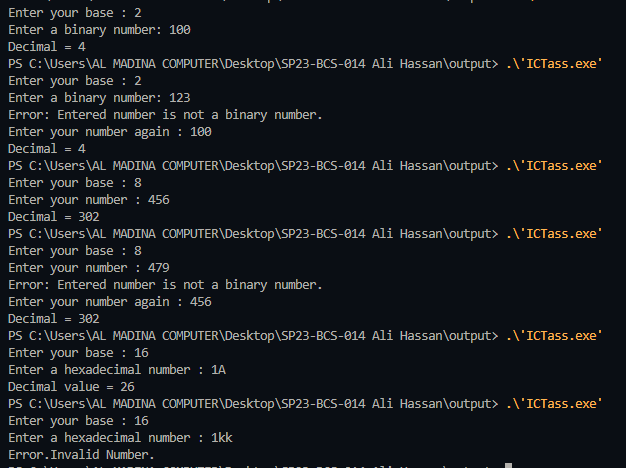
**printf**("Error.Invalid Base number.");

        break;

    }

}

**Output**



# Explanation

**Binary Conversion:**

Binary numbers are represented using only two digits, 0 and 1. To convert a binary number to decimal, the program multiplies each digit by its corresponding power of 2 and adds up the results.

**Here's how the code does it:**

1- The user is prompted to enter a binary number.

2- The program checks if the entered number is valid by ensuring that it contains only 0s and 1s. It does this by repeatedly dividing the number by 10 and checking if the remainder is 0 or 1.

3- If the number is not valid, the program displays an error message and prompts the user to enter the number again.

4- Once a valid binary number is entered, the program converts it to decimal by multiplying each digit by its corresponding power of 2 and adding up the results. This is done using a while loop that repeatedly divides the number by 10, computes the value of the current digit, and adds it to the running total.

5- The final result is displayed as the decimal equivalent of the binary number.

**Octal Conversion:**

Octal numbers are represented using digits 0 to 7. To convert an octal number to decimal, the program multiplies each digit by its corresponding power of 8 and adds up the results.

**Here's how the code does it:**

1-The user is prompted to enter an octal number.

2- The program checks if the entered number is valid by ensuring that it contains only digits between 0 and 7. It does this by repeatedly dividing the number by 10 and checking if each digit is less than 8.

3- If the number is not valid, the program displays an error message and prompts the user to enter the number again.

4- Once a valid octal number is entered, the program converts it to decimal by multiplying each digit by its corresponding power of 8 and adding up the results. This is done using a while loop that repeatedly divides the number by 10, computes the value of the current digit, and adds it to the running total.

5- The final result is displayed as the decimal equivalent of the octal number.

**Hexadecimal Conversion:**

Hexadecimal numbers are represented using digits 0 to 9 and letters A to F (which represent values 10 to 15, respectively). To convert a hexadecimal number to decimal, the program multiplies each digit by its corresponding power of 16 and adds up the results.

**Here's how the code does it:**

1-The user is prompted to enter a hexadecimal number.

2- The program checks if the entered number is valid by ensuring that it contains only digits between 0 and 9 and letters between A and F. It does this by iterating over each character in the input string and checking if it falls within the valid range of characters.

3- If the number is not valid, the program displays an error message and sets the decimal equivalent to 0.

4- Once a valid hexadecimal number is entered, the program converts it to decimal by multiplying each digit by its corresponding power of 16 and adding up the results. This is done using a for loop that iterates over each character in the input string, computes the value of the current digit, and adds it to the running total.

5- The final result is displayed as the decimal equivalent of the hexadecimal number.