মেট্রোপলিটন ইউনিভার্সিটি
Metropolitan University

Project on
Number System Conversion

Course title: Data Structure Lab
Course code: CSE 134

Submitted by,

Sanjida Jahan
ID: 232-115-341
Nilanjana Chowdhury
ID: 232-115-336
Prianka Chowdhury
ID: 232-115-337
Batch: 59th
Section: I
Department of C.S.E.

Submitted to,

Rishad Amin Pulok
Lecturer
Department of C.S.E.

Submission date :12.10.2024

NUMBER SYSTEM CONVERSION

Introduction :

Number system conversion is a fundamental concept in computer science and programming. It involves changing the representation of a number from one base to another, such as converting a decimal number to binary or a hexadecimal number to binary.

In computing, different number systems are used to represent and process data. The most common number systems are binary (base-2), decimal (base-10), octal (base-8), and hexadecimal (base-16). While humans typically use the decimal system, computers work with binary numbers, which are made up of only two digits: 0 and 1. Sometimes, it is necessary to convert numbers from one system to another to make it easier to read or process data.

This project aims to build a program in C that can convert numbers between different number systems, such as binary, decimal, octal, and hexadecimal. The user will be able to input a number in one format and get the corresponding number in another format. The goal is to help users understand how number systems work and to practice using C programming for mathematical calculations and data handling.

In this project we can convert

- **Binary to -**
    1. Decimal
    2. Octal
    3. hexa-decimal
- **Octal to -**
    1. Decimal
    2. Binary
    3. hexa-decimal
- **Decimal to-**

1. Binary
2. Octal
3. hexa-decimal

- **Hexa-decimal to -**
    1. Binary
    2. Decimal
    3. octal

Background study :

In our daily lives, we use the decimal number system (base-10), which consists of 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. However, computers do not use the decimal system to store and process data. Instead, they rely on the binary number system (base-2), which uses only two digits: 0 and 1.

The inspiration for this project comes from understanding how computers work with binary numbers and how humans, who are more comfortable with decimal numbers, need to convert data between different number systems. Additionally, programmers and engineers often use other systems, like octal (base-8) and hexadecimal (base-16), to simplify the work with binary numbers.

How does the program work ?

We define functions for decimal-to-binary, binary-to-decimal, and other mentioned conversions. These functions implement the standard method for the respective conversions.

In the main function, we display a menu of conversion options and ask the user to select one. Depending on the user's choice, we take input and call the corresponding conversion function. The conversion functions perform the conversion and display the result.

Features :

Menu:

```
                                    NUMBER SYSTEM CONVERSION
                                   ================================

*CHOOSE THE CONVERSION*

<< BINARY >>
-------------------------
1: Binary to Decimal.
2: Binary to Octal.
3: Binary to Hexa-Decimal.

<< DECIMAL >>
-------------------------
4: Decimal to Binary.
5: Decimal to Octal.
6: Decimal to Hexa-Decimal.

<< OCTAL >>
--------------------------
7: Octal to Binary.
8: Octal to Decimal.
9: Octal to Hexa-Decimal.

<< HEXA-DECIMAL >>
---------------------------
10: Hexa-Decimal to Binary.
11: Hexa-Decimal to Decimal.
12: Hexa-Decimal to Octal.

ENTER YOUR CHOICE: _
```

1. Binary to Decimal:

```c
//Binary to decimal
long int Bin_to_Dec(long int bin)
{
    int rem,sum=0,i=0;
    while(bin!=0)
    {
        rem=bin%10;
        bin=bin/10;
        sum=sum+rem*pow(2,i);
        i++;
    }

    printf("\nEquivalent Decimal Number : %d",sum);
}
```

```
                                    ***BINARY TO DECIMAL***

Enter the Number in Binary form (0s & 1s): 10001

Equivalent Decimal Number : 17

DO YOU WANT TO CONTINUE = (1/0) :
```

2. Binary to Octal:

```c
long int Bin_to_Oct(long int bin)
{
    int i=0,rem,sum=0,remain[100],len=0;

//Binary to decimal then decimal to octal
    while(bin!=0)
    {
        rem=bin%10;
        bin=bin/10;
        sum=sum+rem*pow(2,i);
        i++;
    }
    i=0;
    while(sum!=0)
    {
        remain[i]=sum%8;
        sum=sum/8;
        i++;
        len++;
    }
    printf("\nEquivalent Octal Number : ");
    for(i=len-1;i>=0;i--)
    {
        printf("%d",remain[i]);
    }
}
```

```
                                ***BINARY TO OCTAL***

Enter the Number in Binary form (0s & 1s): 100010

Equivalent Octal Number : 42

DO YOU WANT TO CONTINUE = (1/0) :
```

3. Binary to Hexa-Decimal:

```
long int Bin_to_Hex(long int bin)
{
    int rem,i=0,sum=0,remain[100],len=0;
//Binary to decimal then decimal to Hexa-decimal
    while(bin!=0)
    {
        rem=bin%10;
        bin=bin/10;
        sum=sum+rem*pow(2,i);
        i++;
    }
    i=0;
    while(sum!=0)
    {
        remain[i]=sum%16;
        sum=sum/16;
        i++;
        len++;
    }
    printf("\nEquivalent Hexa-Decimal Number : ");
    for(i=len-1;i>=0;i--)
    {
        switch(remain[i])
        {
            case 10:
                printf("A"); break;


            case 11:
                printf("B"); break;


            case 12:
                printf("C"); break;



            case 13:
                printf("D"); break;


            case 14:
                printf("E"); break;


            case 15:
                printf("F"); break;


            default:
                printf("%d",remain[i]);
        }


    }
}
```

```
                                ***BINARY TO HEXA-DECIMAL***

Enter the Number in Binary form (0s & 1s): 100111010

Equivalent Hexa-Decimal Number : 13A

DO YOU WANT TO CONTINUE = (1/0) :
```

4. Decimal to Binary:

```c
//Decimal to Binary
long int Dec_to_Bin(long int dec)
{
    int rem[50],i,len=0;
    do
    {
        rem[i]=dec%2;
        dec=dec/2;
        i++;
        len++;
    }
    while(dec!=0);


    printf("\nEquivalent Binary Number : ");
    for(i=len-1;i>=0;i--)
    {
        printf("%d",rem[i]);
    }
}
```

```
                                ***DECIMAL TO BINARY***

Enter the Number in Decimal form (0 to 9): 134

Equivalent Binary Number : 10000110

DO YOU WANT TO CONTINUE = (1/0) :
```

5. Decimal to Octal:

```c
//Decimal to octal
long int Dec_to_Oct(long int dec)
{
    int rem[50],i,len=0;
    do
    {
        rem[i]=dec%8;
        dec=dec/8;
        i++;
        len++;
    }
    while(dec!=0);


    printf("\nEquivalent Octal Number : ");
    for(i=len-1;i>=0;i--)
    {
        printf("%d",rem[i]);
    }
}
```

```
                                ***DECIMAL TO OCTAL***

Enter the Number in Decimal form (0 to 9): 2349

Equivalent Octal Number : 4455

DO YOU WANT TO CONTINUE = (1/0) :
```

6. Decimal to Hexa-decimal:

```
//Decimal to hexa-decimal
long int Dec_to_Hex(long int dec)
{
    int rem[50],i,len=0;
    do
    {
        rem[i]=dec%16;
        dec=dec/16;
        i++;
        len++;
    }
    while(dec!=0);


    printf("\nEquivalent Hexa-Decimal Number : ");
    for(i=len-1;i>=0;i--)
    {
        switch(rem[i])
        {
            case 10:
                printf("A"); break;


            case 11:
                printf("B"); break;


          case 12:
              printf("C"); break;


          case 13:
              printf("D"); break;


          case 14:
              printf("E"); break;


          case 15:
              printf("F"); break;


          default:
              printf("%d",rem[i]);
      }


   }
}
```

```
                                    ***DECIMAL TO HEXA-DECIMAL***

Enter the Number in Decimal form (0 to 9): 12812

Equivalent Hexa-Decimal Number : 320C

DO YOU WANT TO CONTINUE = (1/0) :
```

7. Octal to Binary:

```
long int Oct_to_Bin(long int oct)
{
    int rem[50],len=0,decimal=0,i=0,num,ans;

//Octal to Decimal then Decimal to binary
    while(oct!=0)
    {
        ans=oct % 10;
        decimal = decimal + ans * pow(8,i);
        i++;
        oct = oct/10;
    }


    i=0;
    do
    {
        rem[i]=decimal%2;
        decimal=decimal/2;
        i++;
        len++;
    }
    while(decimal!=0);


    printf("\nEquivalent Binary Number : ");
    for(i=len-1;i>=0;i--)
    {
        printf("%d",rem[i]);
    }
}
```

```
                                ***OCTAL TO BINARY***

Enter the Number in Octal form (0 to 7): 1456

Equivalent Binary Number : 1100101110

DO YOU WANT TO CONTINUE = (1/0) :
```

8. Octal to Decimal:

```
long int Oct_to_Dec(long int oct)
{
    int decimal=0,i=0,num,ans;


    while(oct!=0)
    {
        ans=oct % 10;
        decimal = decimal + ans * pow(8,i);
        i++;
        oct = oct/10;
    }
    printf("\nEquivalent Decimal Number : %d",decimal);
}
```

```
                                ***OCTAL TO DECIMAL***

Enter the Number in Octal form (0 to 7): 753

Equivalent Decimal Number : 491

DO YOU WANT TO CONTINUE = (1/0) :
```

9. Octal to Hexa-decimal:

```c
//Octal to Decimal then Decimal to Hexa-decimal
long int Oct_to_Hex(long int oct)
{
    int rem[50],len=0,decimal=0,i=0,num,ans=0;
    while(oct!=0)
    {
        ans=oct % 10;
        decimal = decimal + ans * pow(8,i);
        i++;
        oct = oct/10;
    }
    i=0;
    while(decimal!=0)
    {
        rem[i]=decimal%16;
        decimal=decimal/16;
        i++;
        len++;
    }
    printf("\nEquivalent Hexa-Decimal Number : ");
    for(i=len-1;i>=0;i--)
    {
        switch(rem[i])
        {
            case 10:
                printf("A"); break;


            case 11:
                printf("B"); break;


            case 12:
                printf("C"); break;


            case 13:
                printf("D"); break;


            case 14:
                printf("E"); break;


            case 15:
                printf("F"); break;


            default:
                printf("%d",rem[i]);
        }


    }
}
```

```
                                ***OCTAL TO HEXA-DECIMAL***

Enter the Number in Octal form (0 to 7): 2347

Equivalent Hexa-Decimal Number : 4E7

DO YOU WANT TO CONTINUE = (1/0) :
```

10. Hexa-decimal to Binary:

```c
//Hexa-decimal to binary
void Hex_to_Bin(char hex[])
{
    int i=0;
    printf("\nEquivalent Binary Number : ");
    for(i=0;i<strlen(hex);i++)
    {
        switch (hex[i])
        {
        case '0':
            printf("0000"); break;
        case '1':
            printf("0001"); break;
        case '2':
            printf("0010"); break;
        case '3':
            printf("0011"); break;
        case '4':
            printf("0100"); break;
        case '5':
            printf("0101"); break;
        case '6':
            printf("0110"); break;
        case '7':
            printf("0111"); break;
        case '8':
            printf("1000"); break;
        case '9':
            printf("1001"); break;

        case 'A':
        case 'a':
            printf("1010"); break;
        case 'B':
        case 'b':
            printf("1011"); break;
        case 'C':
        case 'c':
            printf("1100"); break;
        case 'D':
        case 'd':
            printf("1101"); break;
        case 'E':
        case 'e':
            printf("1110"); break;
        case 'F':
        case 'f':
            printf("1111"); break;


        default:
            printf("\n Invalid hexa digit %c ", hex[i]);
        }
    }

}
```

```
                                ***HEXA-DECIMAL TO BINARY***

Enter the Number in Hexa-Decimal form: A234

Equivalent Binary Number : 1010001000110100

DO YOU WANT TO CONTINUE = (1/0) :
```

11. Hexa-decimal to Decimal:

```
//Hexa-decimal to Decimal
void Hex_to_Dec(char hex[])
{
    int i,num=0,power=0,decimal=0;


    for(i=strlen(hex)-1;i>=0;i--)
    {
        if(hex[i]=='A'||hex[i]=='a')
        {
            num=10;
        }
        else if(hex[i]=='B'||hex[i]=='b')
        {
            num=11;
        }
        else if(hex[i]=='C'||hex[i]=='c')
        {
            num=12;
        }
        else if(hex[i]=='D'||hex[i]=='d')
        {
            num=13;
        }
        else if(hex[i]=='E'||hex[i]=='e')
        {
            num=14;
        }
```

```
        else if(hex[i]=='F'||hex[i]=='f')
        {
            num=15;
        }
        else


        {
            num=hex[i]-48;
        }


        decimal=decimal+num*pow(16,power);
        power++;
    }
    printf("\nEquivalent Decimal Number : %d",decimal);


}
```

```
                                ***HEXA-DECIMAL TO DECIMAL***

Enter the Number in Hexa-Decimal form: 187A

Equivalent Decimal Number : 6266

DO YOU WANT TO CONTINUE = (1/0) :
```

12. Hexa-decimal to Octal:

```c
//hexa-Decimal to octal
void Hex_to_Oct(char hex[])
{
    int i,len,num=0,power=0,decimal=0,rem[100];


    for(i=strlen(hex)-1;i>=0;i--)
    {
        if(hex[i]=='A'||hex[i]=='a')
        {
            num=10;
        }
        else if(hex[i]=='B'||hex[i]=='b')
        {
            num=11;
        }
        else if(hex[i]=='C'||hex[i]=='c')
        {
            num=12;
        }
        else if(hex[i]=='D'||hex[i]=='d')
        {
            num=13;
        }
        else if(hex[i]=='E'||hex[i]=='e')
        {
            num=14;
        }

        else if(hex[i]=='F'||hex[i]=='f')
        {
            num=15;
        }
        else


        {
            num=hex[i]-48;
        }


        decimal=decimal+num*pow(16,power);
        power++;
    }


    i=0,len=0;
    while(decimal!=0)
    {
        rem[i]=decimal%8;
        decimal=decimal/8;
        i++;
        len++;
    }
    printf("\nEquivalent Octal Number : ");
    for(i=len-1;i>=0;i--)
    {
        printf("%d",rem[i]);
    }
```

```
                                ***HEXA-DECIMAL TO DECIMAL***

Enter the Number in Hexa-Decimal form: 187A

Equivalent Decimal Number : 6266

DO YOU WANT TO CONTINUE = (1/0) :
```

Limitations :

In our program, we've identified a few limitations during testing:

- ❖ The program does not support negative numbers.
- ❖ The program does not support fractional numbers.

Future scope :

- ➢ The program could be expanded to include educational features, like step-by-step guides on how the conversions work. This would make it helpful for people who are learning about number systems and computer science ideas.
- ➢ The algorithms used for conversion could be further optimized for better performance, especially when converting large numbers or performing multiple conversions at once.
- ➢ The algorithm could be improved further by allowing the negative and fractional values of a number .

Conclusion:

The goal of this project is to create a C program that can convert numbers between these different number systems. This will allow users to input a number in one system (binary, decimal, octal, or hexadecimal) and convert it to any other system. Understanding these conversions is important for anyone studying computer science and programming.