**Assignment no 3**

**Introduction to computer science**



Submitted by:

**NAME:Asad Rasheed**

**SP23-BCS-029**

**Section-A**

Submitted to:

**Mr. Jawad shafi**

Submitted on: **April 16, 2023**

**Question : build a program that can convert numbers from Binary , Octal and Hexadecimal to decimals?**

Source code

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include<string.h>

/\**write a program to convert binary , octal and hexadecimal number to decimal numbers* \*/

intmain (){

    intnum, digit,base, exponent=0;

    intdecimal=0;

    intvalue=0;

    intlength=0;

    charinput [16];

     printf("enter the choice enter base 2 for binary,8 for octal and 16 for hexadecimal\n");

     scanf("%d",&base);

    switch (base)

    {

    // *program to convert binary to decimal*

    case2:

     printf("enter the  binary number\n");

     scanf("%d",&num);

        // *condition to run the loop until all the digits are added*

    while (num!=0)

    {

         digit=num%10;

            // *checking validity for binary digits*

        if (digit!=0&&digit!=1)

        {

            printf("!!!! Invalid number %d !!!!\n",num);

            printf(" !!! run the program again !!!\n");

            return0;

        }

            // *calculating sum of all digits in decimal*

       decimal+=digit\*pow(base,exponent);

       num/=10;

       exponent++;

    }

        // *printing the decimal number*

    printf(" the decimal equivalent  is %d\n",decimal);

    break;

        // *program to convert octal to decimal*

    case8 :

         printf("enter the  octal number\n");

        scanf("%d",&num);

             // *condition to run the loop until all the digits are added*

        while (num!=0)

        {

            digit=num%10;

            // *checking validity of octal digits*

            if (digit>=8)

            {

                printf(" !!!! Invalid Number %d !!!! \n",num);

                printf(" !!! run the program again !!!\n");

                return0;

            }

            decimal+=digit\*pow (base ,exponent);

            num/=10;

            exponent++;

        }

        printf(" the decimal equivalent  is %d\n",decimal);

        break;

    case16 :

         printf("enter the  hexadecimal number\n");

        scanf("%s",&input);

        intj=0;

        // *to assign whole number to array*

        length=strlen(input);

        length--;

        while ( input [j] !=0)

            {// *checking validity of hexadecimal number*

            if ( input [j]>='0'&&input[j] <='9')

            {// *calculating integers values*

                value=input [j] -'0';

            }

           elseif ( input [j] >='A'&&input[j] <='F' )

            {// *calculating capital alphabets values*

                value=input [j] -'A'+10;

            }

            elseif ( input [j] >='a'&&input [j] <='f' )

            {// *calculating lowercase alphabets values*

                value=input [j] -'a'+10;

            }else {

                printf("!!!! Invalid Number %s!!!!\n",input);

                printf(" !!! run the program again !!!\n");

                return0;

            }

            // *final calculations for hexadecimal values to decimal values*

            decimal+=value\*pow (base,length);

            length--;

            j++;

             }

        printf(" the decimal equivalent  is %d\n",decimal);

         break;

    default:

    printf(" !!! Invalid base !!!\n");

    printf(" !!! enter valid base !!!\n");

    printf(" !!! run the program again !!!\n");

    break;

}

system("pause");

}

**EXPLANATION**

This C program is designed to convert binary, octal, and hexadecimal numbers to their decimal equivalent. The program prompts the user to input the type of base of the number they would like to convert. The user then enters the number of their choice in the respective base. The program then validates the input and performs the conversion to decimal.

The program begins by declaring and initializing several variables, including **num**, **digit**, **base**, **exponent**, **decimal**, **value**, and **length**. The num variable is used to store the user's input number, while the base variable represents the type of base for the number being converted. The exponent variable is used to keep track of the exponent during the conversion process, and the decimal variable stores the final decimal equivalent of the input number. The digit, value, and length variables are used in various steps of the conversion process.

Next, the program prompts the user to enter the base of the number they would like to convert. This is done using the printf and scanf functions to display and receive user input, respectively.

After the user has entered their choice of base, the program uses a switch statement to select the appropriate conversion process for the input number. If the user selects base 2, the program converts the binary number to decimal using a while loop. The loop runs until all the digits of the binary number have been added to the decimal equivalent. Inside the loop, the program validates each digit of the binary number and calculates the decimal equivalent of the number using the pow function. Once the loop has finished running, the program prints the decimal equivalent of the binary number.

If the user selects base 8, the program performs a similar process to convert the octal number to decimal. The loop runs until all the digits of the octal number have been added to the decimal equivalent. Inside the loop, the program validates each digit of the octal number and calculates the decimal equivalent of the number using the pow function. Once the loop has finished running, the program prints the decimal equivalent of the octal number.

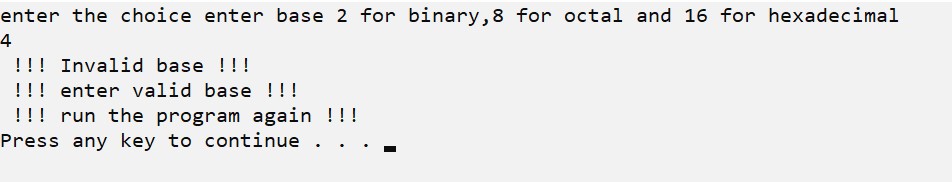
If the user selects base 16, the program converts the hexadecimal number to decimal. The program prompts the user to enter the hexadecimal number as a string using the scanf function. Inside the loop, the program checks the validity of each digit of the hexadecimal number and calculates the decimal equivalent using the pow function. Once the loop has finished running, the program prints the decimal equivalent of the hexadecimal number.

Finally, if the user enters an invalid base, the program displays an error message and prompts the user to run the program again with a valid base.

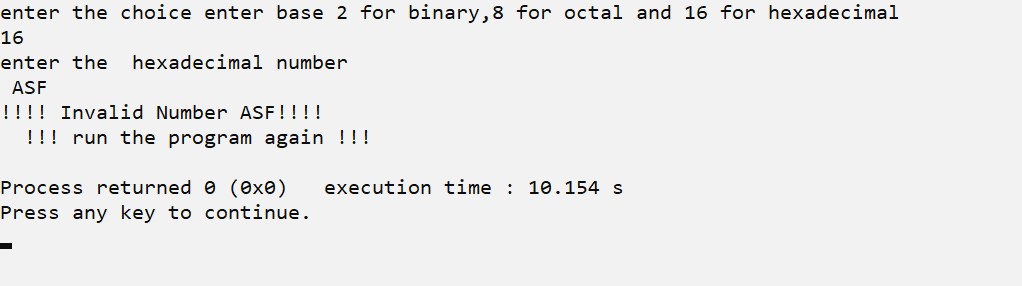
Overall, this program efficiently converts binary, octal, and hexadecimal numbers to decimal using the appropriate conversion formula. The program also includes error handling to ensure that the user enters a valid input for the conversion.

**OUTPUT**

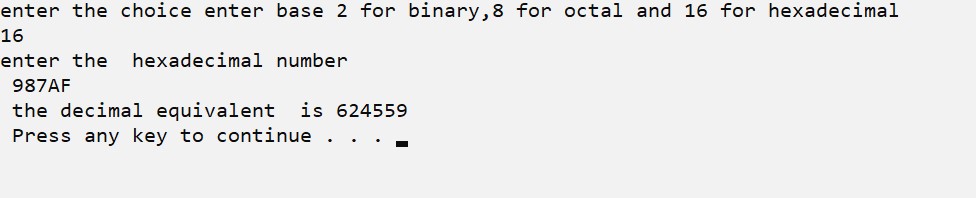
**Validity of base**

****

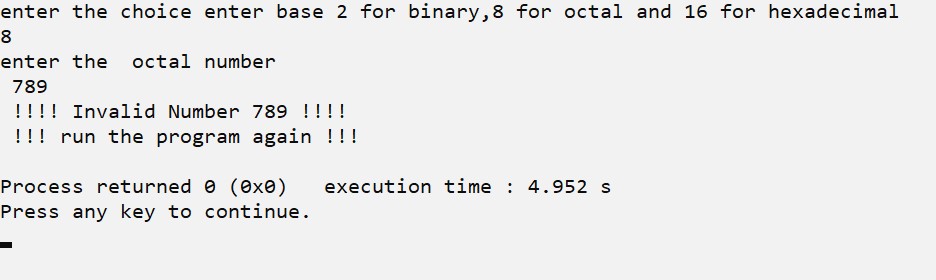
**Validity of hexadecimal number**



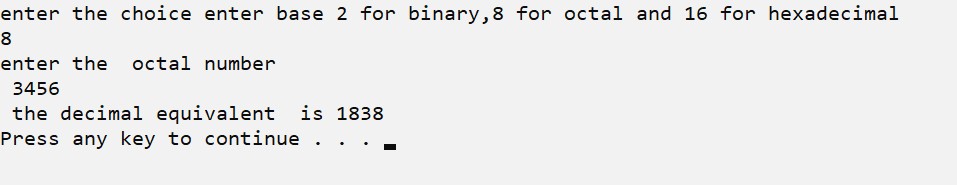
**Conversion from hexadecimal to decimal**



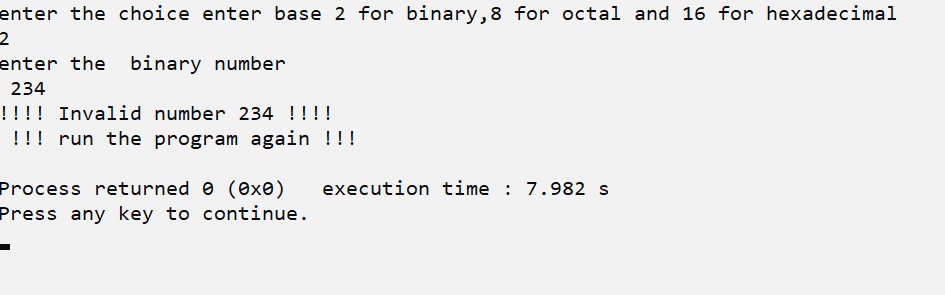
**Validity of Octal number**



**Convertion of Octal to decimal**



**Validity of binary number**



**Conversion of Binary to decimal**

