Date-

Assignment No. :

Problem Statement:

Program in C to convert a number from any base to any base.

Theory:

A number or quantity which is arbitrarily made the fundamental number of a system; also called base 10 is the base number of Decimal system, 2 is the base number of the binary system, 8 is the base number of octal system & 16 is the base number of the hexadecimal system of numeration.

The base of a number system is that number, raised to the zero power, gives the lowest positional value, raised to the 1^{st} power, gives the 2^{nd} positional value and so on. Base conversion is a process to convert a number which can be of binary or octal or decimal or hexadecimal to any of these four bases. To implement this program we have to carefully examine the base of conversion. Complete knowledge of the bases is needed to perform this program. E.g.→

Decimal to Octal conversion→ we can change the base of a decimal number to octal (or base 8) by dividing a given decimal number by 8 repeatedly, until a quotient of 0 is obtained. E.g.-The equivalent octal number of the decimal number 1667 is 3208.

Decimal to Hexadecimal conversion→ we can change the base of a decimal number to hexadecimal (or base 16) by dividing a given decimal number by 16 repeatedly, until a quotient of 0 is obtained. E.g.- The equivalent hexadecimal number of the decimal number 1357 is 54D.

Decimal to Binary conversion→ an easy method of converting a decimal number into a binary number is by dividing the decimal number by 2 respectively, until the quotient of zero is obtained. The binary number is obtained by taking the remainder after each division in the reverse order. The decimal number 52 is equivalent to 110100.

Algorithm:

Input specification: The variables used in the algorithm are listed below with their types and objectives.

Variable	Types	Objective
choice	Character type	Used for checking the condition in do – while loop.
num	Integer type	Number to be entered by the user.
input[ ]	Integer type array	Stores each digit of the input number after converting it to decimal.
ibase	Integer type	Stores the input base.
obase	Integer type	Stores the output base.
c	Integer type	Stores the output.
p	Integer type	Stores the value of i.
hex[ ]	Character type array to store strings	String to be entered by the user.
y[ ]	Integer type	Gives the remainder.
decimal	Integer type	Gives the decimal number.
value	Integer type	Used to calculate decimal.
length	Integer type	Stores the length of the string hex.
i	Integer type	Counter variable used in for loop.

Output specification: Prime factors of X in descending order.

Steps:

Algorithm for method main():

Step 1: [Starting of Do-While loop]
Set i = 1
Step 2: Print “Enter the input base : ”
Step 3: Input ibase
Step 4: If (ibase ≠ 16) Then
Step 5: Print “Enter the number : ”
Step 6: Input num
Step 7: Else
Step 8: Print “Enter the number : ”
Step 9: Input hex
[End of If - Else structure]
Step 10: Print “Enter the output base : ”
Step 11: Input obase
Step 12: If (ibase ≠ 10) Then
Step 13: If (ibase ≠ 16) Then
Step 14: Set c=convert10(num , ibase)
Step 15: Else
Step 16: Set c=convert16to10(hex , ibase)
[End of inner If – Else structure]
Step 17: Else
Step 18: Set c = num
[End of outer If – Else structure]
Step 19: If (obase = 10) Then
Step 20: Print “The output is : ” c
[End of If structure]
Step 21: Repeat through step 22 to step 25 While (c >= 1)
Step 22: Set input[i] = c MOD obase
Step 23: Set c = c/obase

Step 24: Set p = i
Step 25: Set i = i + 1
[End of while loop]
Step 26: If (obase ≠ 10) Then
Step 27: Print “The output is : ”
Step 28: Repeat through step 29 to step 33 For i = p to 1
Step 29: If (input[i] >= 10) Then
Step 30: base16(a , i) //calling of method base16()
Step 31: Else
Step 32: Print input[i]
[End of inner If – Else structure]
Step 33: Set i = i - 1
[End of For loop]
[End of outer If – Else structure]
Step 34: Print “Do you want to continue(Y/N)? :”
Step 35: Input choice
Step 36: If (choice = ’n’ OR choice = ’N’) Then
Step 37: Go to step 39
[End of If structure]
Step 38: Repeat through step 1 to step 37 While (choice=’y’ OR choice=’Y’)
[End of do - while loop]
Step 39: Stop
[End of method main()]

Algorithm for method convert10(n, ibase):

Step 1 : Set i = 0
Step 2 : Set s = 0
Step 3 : Repeat through step 4 to step 7 While (num >= 1)
Step 4 : Set y[i] = num MOD 10
Step 5 : Set num = num / 10
Step 6 : Set p = i
Step 7 : Set i = i + 1
[End of While loop]
Step 8 : Repeat through step 9 to step 10 For i = 0 to p
Step 9 : Set s = s + y[i]*$(ibase)^{i}$

Step 10 : Next i
[End of For loop]
Step 11 : Return s
Step 12 : Stop
[End of method convert10()]

Algorithm for method convert16to10(hex[20], ibase):

Step 1: Set decimal = 0
Step 2: Set i = 0
Step 3: Find the length of the string hex and store it into length
Step 4: Set length = length - 1
Step 5: Repeat through step 6 to step 40 For i = 0 to hex[i]!= NULL
Step 6: If hex[i] = 0 then go to step 7 Else go to step 8
Step 7: i)Set value = 0// case 0
ii) Go to step 38
Step 8: If hex[i] = 1 then go to step 9 Else go to step 10
Step 9: i) Set value = 1// case 1
ii) Go to step 38
Step 10: If hex[i] = 2 then go to step 11 Else go to step 12
Step 11: i) Set value = 2// case 2
ii) Go to step 38
Step 12: If hex[i] = 3 then go to step 13 Else go to step 14
Step 13: i) Set value = 3// case 3
ii) Go to step 38
Step 14: If hex[i] = 4 then go to step 15 Else go to step 16
Step 15: i) Set value = 4// case 4
ii) Go to step 38
Step 16: If hex[i] = 5 then go to step 17 Else go to step 18
Step 17: i) Set value = 5// case 5
ii) Go to step 38
Step 18: If hex[i] = 6 then go to step 19 Else go to step 20
Step 19: i) Set value = 6// case 6
ii) Go to step 38
Step 20: If hex[i] = 7 then go to step 21 Else go to step 22
Step 21: i) Set value = 7// case 7
ii) Go to step 38
Step 22: If hex[i] = 8 then go to step 23 Else go to step 24

Step 23: i) Set value = 8// case 8
ii) Go to step 38
Step 24: If hex[i] = 9 then go to step 25 Else go to step 26
Step 25: i) Set value = 9// case 9
ii) Go to step 38
Step 26: If hex[i] = 10 then go to step 27 Else go to step 28
Step 27: i) Set value = 10// case A and case a
ii) Go to step 38
Step 28: If hex[i] = 11 then go to step 29 Else go to step 30
Step 29: i) Set value = 11// case B and case b
ii) Go to step 38
Step 30: If hex[i] = 12 then go to step 31 Else go to step 32
Step 31: i) Set value = 12// case C and case c
ii) Go to step 38
Step 32: If hex[i] = 13 then go to step 33 Else go to step 34
Step 33: i) Set value = 13// case D and case d
ii) Go to step 38
Step 34: If hex[i] = 14 then go to step 35 Else go to step 36
Step 35: i) Set value = 14// case E and case e
ii) Go to step 38
Step 36: If hex[i] = 15 then go to step 37 Else go to step 38
Step 37: i) Set value = 15// case F and case f
ii) Go to step 38
[End of If-Else block]
Step 38: Set decimal = decimal + value*$(ibase)^{length}$
Step 39: Set length = length - 1
Step 40: Next i
[End of For loop]
Step 41: Return decimal
Step 42: Stop
[End of method convert16to10()]

Algorithm for method base16(input[25], i):

Step 1: Set c = input[i] + 55
Step 2: Print c
Step 3: Stop
[End of method base16()]

Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<math.h>
long int convert10(long int,long int);
void base16(long int input[25],long int);
long int convert16to10(char hex[20],long int);
int l;

int main()
{
	char ch;
	do //continuity checking loop
	{
		long int num,input[25],ib,ob,c,p,i=1;
		char hex[17]; //for hexadecimal inputs
		printf("Enter the input base: ");
		scanf("%ld",&ib); //input base scanning
		if(ib!=16) //when input is not hexadecimal
		{
			printf("Enter the number : ");
			scanf("%ld",&num);
		}
		else //when input is hexadecimal
		{
			printf("Enter the number : ");
			fflush(stdin);
```

```
        gets(hex); //storing the hexadecimal input to the array
}
printf("Enter the output base : ");
scanf("%ld",&ob); //scanning the output base
if(ib!=10) //when input is not decimal
{
        if(ib!=16) /*when input base is not 16 but output base is
                        10*/
        c=convert10(num,ib);
        else //when input base is 16 and ouput base is 10
        c=convert16to10(hex,ib);
}
else //when input is not decimal
c=num; //setting the input in c
if(ob==10) //when ouput base is 10
{
        printf("The output is : %ld",c); /*when input and output
                                                base are same*/
}
while(c>=1)
{
        input[i]=c%ob; /*storing the mod value of input and
                                output base*/
        c=c/ob; //storing the remainder in c
        p=i; //storing the value value of i
        i++;
}
if(ob!=10) //when output base not equals to 10
{
        printf("The output is : ");
```

```
                for(i=p;i>=1;i--) //printing output
                {
                        if(input[i]>=10)
                        base16(input,i);
                        else
                        printf("%ld",input[i]);
                }
        }
        printf("\nDo you want to continue(Y/N)? : "); /*continuity
                                                        check*/
        fflush(stdin);
        scanf("%c",&ch);
        if(ch=='n' || ch=='N')
        return 0;
    }while(ch=='y' || ch=='Y');
return 0;
}
long int convert10(long int num,long int ibase)//Function for converting
                                                //from any base to base 10
{
    long int y[25],i=0,s=0,p,decimal,value;
    while(num>=1) //conversion part
    {
        y[i]=num%10;
        num=num/10;
        p=i;
        i++;
    }
    for(i=0;i<=p;i++)
    s=s+y[i]*pow(ibase,i);
```

```
        return s;
}
long int convert16to10(char hex[20],long int ibase)
{
        long int length,value,decimal=0;
        int i=0;
        length=strlen(hex);
        length--;
        for(i=0;hex[i]!='\0';i++)
        {
                switch(hex[i])
                {
                        case '0': value=0;
                        break;
                        case '1': value=1;
                        break;
                        case '2': value=2;
                        break;
                        case '3': value=3;
                        break;
                        case '4': value=4;
                        break;
                        case '5': value=5;
                        break;
                        case '6': value=6;
                        break;
                        case '7': value=7;
                        break;
                        case '8': value=8;
                        break;
```

```
                case '9': value=9;
                break;
                case 'A':
                case 'a': value=10;
                break;
                case 'B':
                case 'b': value=11;
                break;
                case 'C':
                case 'c': value=12;
                break;
                case 'D':
                case 'd': value=13;
                break;
                case 'E':
                case 'e': value=14;
                break;

                case 'F':
                case 'f': value=15;
                break;
            }
            decimal=decimal+value*pow(ibase,length);
            length--;
        }
        return decimal;
}
```

```
void base16(long int input[25], long int i)
{
	char c;
	c=input[i]+55;
	printf("%c",c);
}
```

Input & Output:

Set 1:

```
Enter the input base: 10
Enter the number : 61
Enter the output base : 2
The output is : 111101
Do you want to continue(Y/N)? :
```

Set 2:

```
Enter the input base: 10
Enter the number : 90
Enter the output base : 16
The output is : 5A
Do you want to continue(Y/N)? :
```

Set 3:

```
Enter the input base: 2
Enter the number : 100111
Enter the output base : 10
The output is : 39
Do you want to continue(Y/N)? :
```

Set 4:

```
Enter the input base: 10
Enter the number : 78
Enter the output base : 8
The output is : 116
Do you want to continue(Y/N)? :
```

Discussion:

1. To convert a number from one base to another one must enter two different bases.
2. If the entered bases are same then the output will be same as like the input. So for better programming one must check at first whether the bases are same or not.
3. If one enters the negative number for bases and also for number of that corresponding base, then user cannot get the desired output. So to get the desired output one must enter the positive number for bases and for numbers.