**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 1st power, gives the 2nd 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():

1. [Starting of Do-While loop] Set i = 1
2. Print “Enter the input base : ”
3. Input ibase
4. If ( ibase ≠ 16 ) Then
5. Print “Enter the number : ”
6. Input num
7. Else
8. Print “Enter the number : ”
9. Input hex

[End of If - Else structure]

1. Print “Enter the output base : ”
2. Input obase
3. If ( ibase ≠ 10 ) Then
4. If ( ibase ≠ 16 ) Then
5. Set c=convert10(num , ibase)
6. Else
7. Set c=convert16to10(hex , ibase)

[End of inner If – Else structure]

1. Else
2. Set c = num

[End of outer If – Else structure]

1. If (obase = 10) Then
2. Print “The output is : ” c

[End of If structure]

1. Repeat through step 22 to step 25 While (c >= 1)
2. Set input[i] = c MOD obase
3. Set c = c/obase
4. Set p = i
5. Set i = i + 1

[End of while loop]

1. If (obase ≠ 10) Then
2. Print “The output is : ”
3. Repeat through step 29 to step 33 For i = p to 1
4. If ( input[i] >= 10 ) Then
5. base16(a , i) //calling of method base16()
6. Else
7. Print input[i]

[End of inner If – Else structure]

1. Set i = i - 1

[End of For loop]

[End of outer If – Else structure]

1. Print “Do you want to continue(Y/N)? :”
2. Input choice
3. If (choice = ’n’ OR choice = ’N’) Then
4. Go to step 39

[End of If structure]

1. Repeat through step 1 to step 37 While (choice=’y’ OR choice=’Y’)

[End of do - while loop]

1. Stop

[End of method main()]

Algorithm for method convert10(n, ibase):

1. Set i = 0
2. Set s = 0
3. Repeat through step 4 to step 7 While (num >= 1)
4. Set y[i] = num MOD 10
5. Set num = num / 10
6. Set p = i
7. Set i = i + 1

[End of While loop]

1. Repeat through step 9 to step 10 For i = 0 to p
2. Set s = s + y[i]\*(ibase)**i**
3. Next i

[End of For loop]

1. Return s
2. Stop [End of method convert10()]

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

1. Set decimal = 0
2. Set i = 0
3. Find the length of the string hex and store it into length
4. Set length = length - 1
5. Repeat through step 6 to step 40 For i = 0 to hex[i]!= NULL
6. If hex[i] = 0 then go to step 7 Else go to step 8
7. i)Set value = 0// case 0

ii) Go to step 38

1. If hex[i] = 1 then go to step 9 Else go to step 10
2. i) Set value = 1// case 1

ii) Go to step 38

1. If hex[i] = 2 then go to step 11 Else go to step 12
2. i) Set value = 2// case 2

ii) Go to step 38

1. If hex[i] = 3 then go to step 13 Else go to step 14
2. i) Set value = 3// case 3

ii) Go to step 38

1. If hex[i] = 4 then go to step 15 Else go to step 16
2. i) Set value = 4// case 4

ii) Go to step 38

1. If hex[i] = 5 then go to step 17 Else go to step 18
2. i) Set value = 5// case 5

ii) Go to step 38

1. If hex[i] = 6 then go to step 19 Else go to step 20
2. i) Set value = 6// case 6

ii) Go to step 38

1. If hex[i] = 7 then go to step 21 Else go to step 22
2. i) Set value = 7// case 7

ii) Go to step 38

1. If hex[i] = 8 then go to step 23 Else go to step 24
2. i) Set value = 8// case 8

ii) Go to step 38

1. If hex[i] = 9 then go to step 25 Else go to step 26
2. i) Set value = 9// case 9

ii) Go to step 38

1. If hex[i] = 10 then go to step 27 Else go to step 28
2. i) Set value = 10// case A and case a

ii) Go to step 38

1. If hex[i] = 11 then go to step 29 Else go to step 30
2. i) Set value = 11// case B and case b

ii) Go to step 38

1. If hex[i] = 12 then go to step 31 Else go to step 32
2. i) Set value = 12// case C and case c

ii) Go to step 38

1. If hex[i] = 13 then go to step 33 Else go to step 34
2. i) Set value = 13// case D and case d

ii) Go to step 38

1. If hex[i] = 14 then go to step 35 Else go to step 36
2. i) Set value = 14// case E and case e

ii) Go to step 38

1. If hex[i] = 15 then go to step 37 Else go to step 38
2. i) Set value = 15// case F and case f

ii) Go to step 38

[End of If-Else block]

1. Set decimal = decimal + value\*(ibase)length
2. Set length = length - 1
3. Next i

[End of For loop]

1. Return decimal
2. Stop

[End of method convert16to10()]

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

1. Set c = input[i] + 55
2. Print c
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;

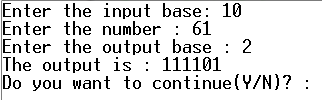
c=input[i]+55;

printf("%c",c);

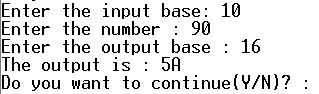
}

**Input & Output:**

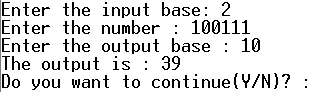
Set 1:



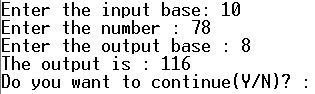
Set 2:



Set 3:



Set 4:



**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.