**SIKSHA ‘O’ANUSANDHAN**

## DEEMED TO BE UNIVERSITY

**Admission Batch: 2020 Session: 2020-2021**

Project Report

# Programming and Data Structures using C (CA3001)

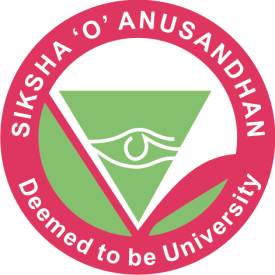
***Submitted by***

Name: Gopal Krushna Padhi

Registration No.: 2061020073

Branch: MCA

Semester: 1st Section: ‘B’



**Department of Computer Science & Engineering Faculty of Engineering & Technology (ITER)**

**Jagamohan Nagar, Jagamara, Bhubaneswar, Odisha – 751030**

|  |  |
| --- | --- |
| Ex.No.**17**  **16.03.2021** | **PROJECT REPORT - SUMMARY** |

|  |
| --- |
| **AIM:** |

This Mini Project for Number System Conversion, the client can effectively change over any number framework such as: binary to decimal, binary to octal, binary to hexadecimal. Decimal to binary, Decimal to octal and Decimal to hexadecimal. Octal to binary, Octal to decimal and Octal to hexadecimal. Hexadecimal to binary, Hexadecimal to decimal and hexadecimal to octal. The client must select numbers and after that enter the number agreeing to their change. This extend has made the number framework transformation quick and simple for the clients.

**Number Conversion**

**Binary to Decimal:-**

Step1-START

Step2-Read binary number

Step3-While(check!=0)

Num=check%10;

If(num>1)

Display is not a binary number

Else

Check=check/10;

Step4-Read remain,sum;

Step5-While(bin!=0)

Rem=bin%10;

Bin=bin/10;

Sum=sum+Rem\*pow(2,i);

I++

Step6-Display the decimal number

Step7-STOP

**Binary To Octal:-**

Step1- START

Step2- Read binary number

Step3-While(check!=0)

Num=check%10;

If(num>1)

Display is not a binary number

Else

Check=check/10;

Step4-Read I,Rem;Sum,Remain,sen

Step5- While(bin!=0)

Rem=bin%10;

Bin=bin/10;

Sum=sum+Rem\*pow(2,i);

I++

Step6-While(sum!=0)

Remain[i]=sum%&;

Sum=sum/&;

I++;

Len++;

Step7-Display the octal number

Step8-STOP

**Binary to Hexadecimal:-**

Step1- START

Step2- Read binary number

Step3- While(check!=0)

Num=check%10;

If(num>1)

Display is not a binary number

Else

Check=check/10;

Step4-Same as binary to octal

Step5- While(bin!=0)

Rem=bin%10;

Bin=bin/10;

Sum=sum+Rem\*pow(2,i);

I++

Step6-Remain[i]=Sum%16

Sum=sum/16;

I++;

Ren++;

Step7-Display Hexadecimal number

Step8-STOP

**Decimal To Binary**

Step1-START

Step2-Read decimal number

Step3-Read rem,I,Sen

Step4-do

Rem[i]=dec%2;

Dec=dec/2;

I++;

Sen++;

While(dec!=0)

Step5-Display Decimal number

Step6-STOP

**Decimal to octal**

Step1-START

Step2-Read decimal number

Step3-Read rem,I,sen

Step4-do

Rem[i]=dec%8;

Dec=dec/8;

I++;

Sen++;

While(dec!=0)

Step5-Display octal number

Step6-STOP

**Decimal to Hexadecimal**

Step1- START

Step2-Read Decimal number

Step3-Read rem,I,sen

Step4- Rem[i]=dec%16;

Dec=dec/16;

I++;

Sen++;

While(dec!=0)

Step5-Display Hexadecimal

Step6-STOP

**octal to binary**

Step1-START

Step2-Read the octal number

Step3-Read rem,sen,decimal,I,num,ans

Step4-While(oct!=0)

Ans=oct%10

Decimal=Decimal+ans\*pow(&,i);

I++;

Oct=oct/10;

Step5-i=0

Do

Rem[i]=decimal%2;

Decimal=decimal/2

I++;

Sen++;

Stept6-Display Binary number

Step7-STOP

**octal to Decimal**

Step1-START

Step2-Read the octal number

Step3-Check=oct

While(check!=0)

If(num>7)

Display is not octal number

Else

Check=check/10

I++

Step4-Read Decimal,I,num,ans

Step5-while(oct!=0)

Ans=oct%10

Decimal=decimal+ans\*pow/8,i);

I++

Oct=oct/10

Stept6-Display decimal number

Step7-STOP

**octal to hexadecimal**

Step1-START

Step2-Read octal number

Step3-check=oct

While(check!=0)

If(num>7)

Display is not octal number

Else

Check=check/10

I++

Step4-Read Decimal,I,num,ans

Step5- while(oct!=0)

Ans=oct%10

Decimal=decimal+ans\*pow/8,i);

I++

Oct=oct/10

Step6-i=0;

While(decimal!=0\_)

Rem[i]=decimal%16;

Decimal=decimal/16

Step7-Display Hexadecimal

Step8-STOP

**Hexadecimal to Binary**

Step1-START

Step2-Read hexadecimal number

Step3-for(i=strsen(hex)-1;i>0;i++)

If(hex[i]>’f’ && hex[i]<=’z’ ||dfhex[i]>’F’&&hex[i]<=’z’)

Step4-INTI=0

Step5-Display binary number

For(i=0;i<strsen(hex);i++)

Switch(hex[i])

Display case(s);

Step6-Exit

**Hexadecimal to decimal**

Step1-START

Step2-Read the hexadecimal

Step3- for(i=strsen(hex)-1;i>0;i++)

If(hex[i]>’f’ && hex[i]<=’z’ ||dfhex[i]>’F’&&hex[i]<=’z’)

Step4-Read I,num,power,decimal

Step5-For(i=strsen(hex)-1;i>=0;i--)

Decimal+num\*pow(16,power);

Power++

Step6-Display Decimal number

Step7-EXIT

**Hexadecimal to octal**

Step1-START

Step2-Read hexadecimal number

Step3- for(i=strsen(hex)-1;i>0;i++)

If(hex[i]>’f’ && hex[i]<=’z’ ||dfhex[i]>’F’&&hex[i]<=’z’)

Step4-Read I,sen,num,power,decimal,rem[100]

Step5- For(i=strsen(hex)-1;i>=0;i--)

Decimal+num\*pow(16,power);

Power++

Step6-i=o,sen=0

While(decimal!=0)

Rem[i]=decimal%8;

Decimal=decimal/8;

I++;

Sen++;

Step7-Display the octal number

Step8-STOP

|  |
| --- |
| **RESULT:** |

The Number System Conversion Project in C is a consoled based application and created using c programming language. This simple mini project for Number System Conversion is complete and totally error free.

|  |  |
| --- | --- |
| Ex.No.**18**  **16.03.2021** | **PROJECT REPORT - IMPLEMENTATION** |

**Binary to Decimal:-**

printf("\n\*\*BINARY TO DECIMAL\*\*\n");

D:

printf("\nEnter the Number in Binary form (0s & 1s): ");

scanf("%ld",&bin);

// CHECKING INPUT IS IN BINARY FORM

check=bin;

while(check!=0)

{

num=check%10;

if(num>1)

{

printf("\n%d IS NOT BINARY NUMBER.\n",bin);

printf("\*\*TRY AGAIN\*\n");

goto D;

}

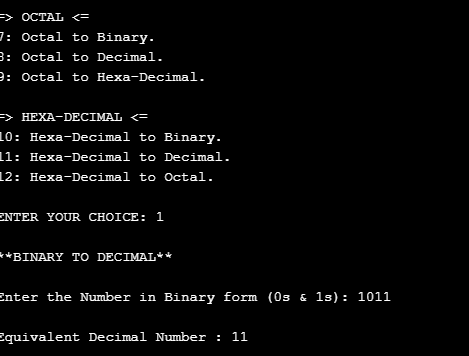
else

check=check/10;

}

Bin\_to\_Dec(bin); break;

**Output:-**



**Binary To Octal:-**

printf("\n\*\*BINARY TO OCTAL\*\*\n");

E:

printf("\nEnter the Number in Binary form (0s & 1s): ");

scanf("%ld",&bin);

// CHECKING INPUT IS IN BINARY FORM

check=bin;

while(check!=0)

{

num=check%10;

if(num>1)

{

printf("\n%d IS NOT BINARY NUMBER.\n",bin);

printf("\*\*TRY AGAIN\*\n");

goto E;

}

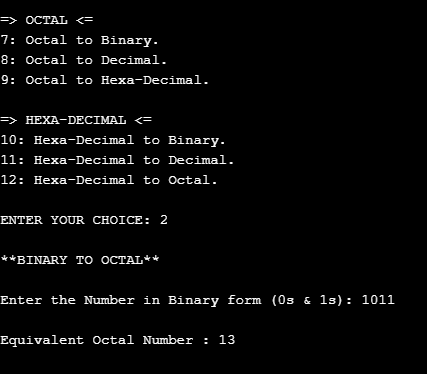
else

check=check/10;

}

Bin\_to\_Oct(bin); break;

**Output:-**



**Binary to Hexadecimal:-**

printf("\n\*\*BINARY TO HEXA-DECIMAL\*\*\n");

F:

printf("\nEnter the Number in Binary form (0s & 1s): ");

scanf("%ld",&bin);

// CHECKING INPUT IS IN BINARY FORM

check=bin;

while(check!=0)

{

num=check%10;

if(num>1)

{

printf("\n%d IS NOT BINARY NUMBER.\n",bin);

printf("\*\*TRY AGAIN\*\n");

goto F;

}

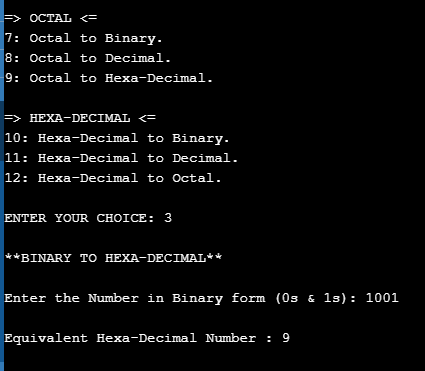
else

check=check/10;

}

Bin\_to\_Hex(bin); break;

**Output:-**



**Decimal To Binary**

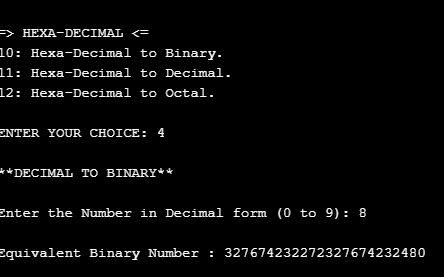
printf("\n\*\*DECIMAL TO BINARY\*\*\n");

printf("\nEnter the Number in Decimal form (0 to 9): ");

scanf("%ld",&dec);

Dec\_to\_Bin(dec); break;

**Output:-**



**Decimal to octal**

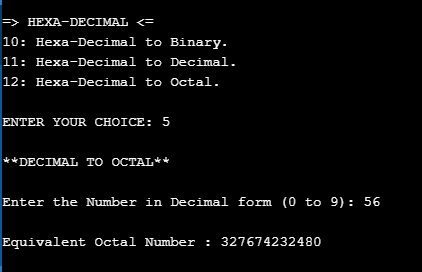
printf("\n\*\*DECIMAL TO OCTAL\*\*\n");

printf("\nEnter the Number in Decimal form (0 to 9): ");

scanf("%ld",&dec);

Dec\_to\_Oct(dec); break;

**Output:-**



**Decimal to Hexadecimal**

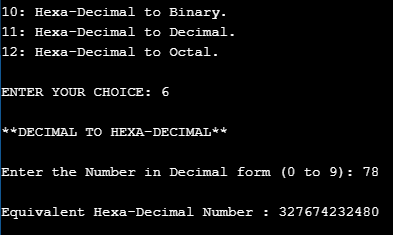
printf("\n\*\*DECIMAL TO HEXA-DECIMAL\*\*\n");

printf("\nEnter the Number in Decimal form (0 to 9): ");

scanf("%ld",&dec);

Dec\_to\_Hex(dec); break;

**Output:-**



**octal to binary**

printf("\n\*\*OCTAL TO BINARY\*\*\n");

A:

printf("\nEnter the Number in Octal form (0 to 7): ");

scanf("%ld",&oct);

// CHECKING INPUT IS IN OCTAL FORM

check=oct;

while(check!=0)

{

num=check%10;

if(num>7)

{

printf("\n%d IS NOT OCTAL NUMBER.\n",num);

goto A;

}

else

{

check=check/10;

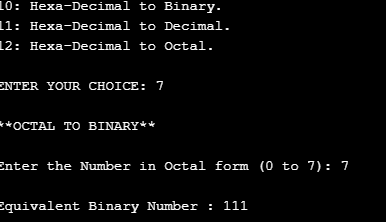
i++;

}

}

Oct\_to\_Bin(oct); break;

**Output:-**



**octal to Decimal**

printf("\n\*\*OCTAL TO DECIMAL\*\*\n");

B:

printf("\nEnter the Number in Octal form (0 to 7): ");

scanf("%ld",&oct);

// CHECKING INPUT IS IN OCTAL FORM

check=oct;

while(check!=0)

{

num=check%10;

if(num>7)

{

printf("\n%d IS NOT OCTAL NUMBER.\n",num);

goto B;

}

else

{

check=check/10;

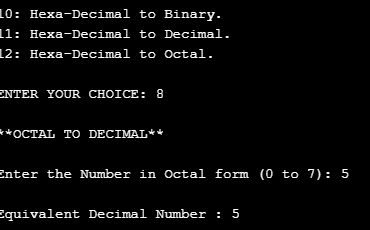
i++;

}

}

Oct\_to\_Dec(oct); break;

**Output:-**



**octal to hexadecimal**

printf("\n\*\*OCTAL TO HEXA-DECIMAL\*\*\n");

C:

printf("\nEnter the Number in Octal form (0 to 7): ");

scanf("%ld",&oct);

// CHECKING INPUT IS IN OCTAL FORM

check=oct;

while(check!=0)

{

num=check%10;

if(num>7)

{

printf("\n%d IS NOT OCTAL NUMBER.\n",num);

goto C;

}

else

{

check=check/10;

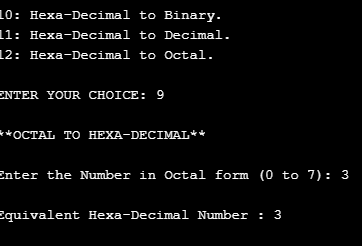
i++;

}

}

Oct\_to\_Hex(oct); break;

**Output:-**



**Hexadecimal to Binary**

printf("\n\*\*HEXA-DECIMAL TO BINARY\*\*\n");

X:

printf("\nEnter the Number in Hexa-Decimal form: ");

scanf("%s",&hex);

//check

for(i=strlen(hex)-1;i>=0;i--)

{

if(hex[i]>'f' && hex[i]<='z' || hex[i]>'F'&& hex[i]<='Z')

{

printf("\nYou have to Enter Hexa-Decimal Number.\n");

printf("'%c' IS NOT Hexa-Decimal Number.\n",hex[i]);

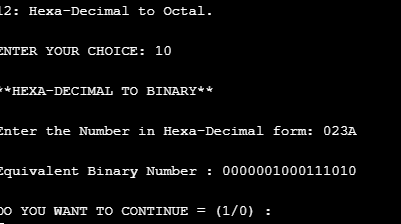
goto X;

}

}

Hex\_to\_Bin(hex); break;

**Output:-**



**Hexadecimal to decimal**

printf("\n\*\*HEXA-DECIMAL TO DECIMAL\*\*\n");

Y:

printf("\nEnter the Number in Hexa-Decimal form: ");

scanf("%s",&hex);

//check

for(i=strlen(hex)-1;i>=0;i--)

{

if(hex[i]>'f' && hex[i]<='z' || hex[i]>'F'&& hex[i]<='Z')

{

printf("\nYou have to Enter Hexa-Decimal Number.\n");

printf("'%c' IS NOT Hexa-Decimal Number.\n",hex[i]);

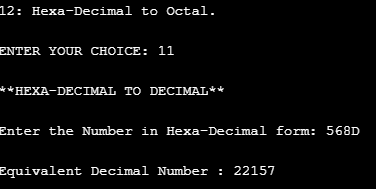
goto Y;

}

}

Hex\_to\_Dec(hex); break;

**Output:-**



**Hexadecimal to octal**

printf("\n\*\*HEXA-DECIMAL TO OCTAL\*\*\n");

Z:

printf("\nEnter the Number in Hexa-Decimal form: ");

scanf("%s",&hex);

//check

for(i=strlen(hex)-1;i>=0;i--)

{

if(hex[i]>'f' && hex[i]<='z' || hex[i]>'F'&& hex[i]<='Z')

{

printf("\nYou have to Enter Hexa-Decimal Number.\n");

printf("'%c' IS NOT Hexa-Decimal Number.\n",hex[i]);

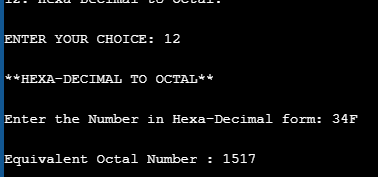
goto Z;

}

}

Hex\_to\_Oct(hex); break;

**Output:-**



|  |
| --- |
| **RESULT:** |

Thus, the processes Binary to Decimal, Binary To Octal, Binary to Hexadecimal, Decimal To Binary, Decimal to octal, Decimal to Hexadecimal, octal to binary, octal to Decimal, octal to hexadecimal, Hexadecimal to Binary, Hexadecimal to decimal, Hexadecimal to octalfor Number conversion. Application has been implemented and the output has been verified successfully.

Program:-

#include <stdio.h>

#include <math.h>

#include<string.h>

#include <conio.h>

long int Bin\_to\_Dec(long int); //1:BINARY TO DECIMAL

long int Bin\_to\_Oct(long int); //2:BINARY TO OCTAL

long int Bin\_to\_Hex(long int); //3:BINARY TO HEXA-DECIMAL

long int Dec\_to\_Bin(long int); //4:DECIMAL TO BINARY

long int Dec\_to\_Oct(long int); //5:DECIMAL TO OCTAL

long int Dec\_to\_Hex(long int); //6:DECIMAL TO HEXA-DECIMAL

long int Oct\_to\_Bin(long int); //7:OCTAL TO BINARY

long int Oct\_to\_Dec(long int); //8:OCTAL TO DECIMAL

long int Oct\_to\_Hex(long int); //9:OCTAL TO HEXA-DECIMAL

void Hex\_to\_Bin(char []); //10:HEXA-DECIMAL TO BINARY

void Hex\_to\_Dec(char []); //11:HEXA-DECIMAL TO DECIMAL

void Hex\_to\_Oct(char []); //12:HEXA-DECIMAL TO OCTAL

int main()

{

int op,num=1,check;

long int bin,oct,dec;

char hex[100];

int i,j,space; // FOR PATTERN

printf("\t\tWELCOME TO NUMBER SYSTEM CONVERSION\n\n");

while(num!=0)

{

printf("\t\t>>>>>> CHOOSE THE CONVERSION <<<<<<\n\n");

printf("=> BINARY <=\n");

printf("1: Binary to Decimal.\n2: Binary to Octal.\n3: Binary to Hexa-Decimal.\n");

printf("\n=> DECIMAL <=\n");

printf("4: Decimal to Binary.\n5: Decimal to Octal.\n6: Decimal to Hexa-Decimal.\n");

printf("\n=> OCTAL <=\n");

printf("7: Octal to Binary.\n8: Octal to Decimal.\n9: Octal to Hexa-Decimal.\n");

printf("\n=> HEXA-DECIMAL <=\n");

printf("10: Hexa-Decimal to Binary.\n11: Hexa-Decimal to Decimal.\n12: Hexa-Decimal to Octal.\n");

printf("\nENTER YOUR CHOICE: ");

scanf("%d",&op);

switch(op)

{

case 1:

printf("\n\*\*BINARY TO DECIMAL\*\*\n");

D:

printf("\nEnter the Number in Binary form (0s & 1s): ");

scanf("%ld",&bin);

// CHECKING INPUT IS IN BINARY FORM

check=bin;

while(check!=0)

{

num=check%10;

if(num>1)

{

printf("\n%d IS NOT BINARY NUMBER.\n",bin);

printf("\*\*TRY AGAIN\*\n");

goto D;

}

else

check=check/10;

}

Bin\_to\_Dec(bin); break;

case 2:

printf("\n\*\*BINARY TO OCTAL\*\*\n");

E:

printf("\nEnter the Number in Binary form (0s & 1s): ");

scanf("%ld",&bin);

// CHECKING INPUT IS IN BINARY FORM

check=bin;

while(check!=0)

{

num=check%10;

if(num>1)

{

printf("\n%d IS NOT BINARY NUMBER.\n",bin);

printf("\*\*TRY AGAIN\*\n");

goto E;

}

else

check=check/10;

}

Bin\_to\_Oct(bin); break;

case 3:

printf("\n\*\*BINARY TO HEXA-DECIMAL\*\*\n");

F:

printf("\nEnter the Number in Binary form (0s & 1s): ");

scanf("%ld",&bin);

// CHECKING INPUT IS IN BINARY FORM

check=bin;

while(check!=0)

{

num=check%10;

if(num>1)

{

printf("\n%d IS NOT BINARY NUMBER.\n",bin);

printf("\*\*TRY AGAIN\*\n");

goto F;

}

else

check=check/10;

}

Bin\_to\_Hex(bin); break;

case 4:

printf("\n\*\*DECIMAL TO BINARY\*\*\n");

printf("\nEnter the Number in Decimal form (0 to 9): ");

scanf("%ld",&dec);

Dec\_to\_Bin(dec); break;

case 5:

printf("\n\*\*DECIMAL TO OCTAL\*\*\n");

printf("\nEnter the Number in Decimal form (0 to 9): ");

scanf("%ld",&dec);

Dec\_to\_Oct(dec); break;

case 6:

printf("\n\*\*DECIMAL TO HEXA-DECIMAL\*\*\n");

printf("\nEnter the Number in Decimal form (0 to 9): ");

scanf("%ld",&dec);

Dec\_to\_Hex(dec); break;

case 7:

printf("\n\*\*OCTAL TO BINARY\*\*\n");

A:

printf("\nEnter the Number in Octal form (0 to 7): ");

scanf("%ld",&oct);

// CHECKING INPUT IS IN OCTAL FORM

check=oct;

while(check!=0)

{

num=check%10;

if(num>7)

{

printf("\n%d IS NOT OCTAL NUMBER.\n",num);

goto A;

}

else

{

check=check/10;

i++;

}

}

Oct\_to\_Bin(oct); break;

case 8:

printf("\n\*\*OCTAL TO DECIMAL\*\*\n");

B:

printf("\nEnter the Number in Octal form (0 to 7): ");

scanf("%ld",&oct);

// CHECKING INPUT IS IN OCTAL FORM

check=oct;

while(check!=0)

{

num=check%10;

if(num>7)

{

printf("\n%d IS NOT OCTAL NUMBER.\n",num);

goto B;

}

else

{

check=check/10;

i++;

}

}

Oct\_to\_Dec(oct); break;

case 9:

printf("\n\*\*OCTAL TO HEXA-DECIMAL\*\*\n");

C:

printf("\nEnter the Number in Octal form (0 to 7): ");

scanf("%ld",&oct);

// CHECKING INPUT IS IN OCTAL FORM

check=oct;

while(check!=0)

{

num=check%10;

if(num>7)

{

printf("\n%d IS NOT OCTAL NUMBER.\n",num);

goto C;

}

else

{

check=check/10;

i++;

}

}

Oct\_to\_Hex(oct); break;

case 10:

printf("\n\*\*HEXA-DECIMAL TO BINARY\*\*\n");

X:

printf("\nEnter the Number in Hexa-Decimal form: ");

scanf("%s",&hex);

//check

for(i=strlen(hex)-1;i>=0;i--)

{

if(hex[i]>'f' && hex[i]<='z' || hex[i]>'F'&& hex[i]<='Z')

{

printf("\nYou have to Enter Hexa-Decimal Number.\n");

printf("'%c' IS NOT Hexa-Decimal Number.\n",hex[i]);

goto X;

}

}

Hex\_to\_Bin(hex); break;

case 11:

printf("\n\*\*HEXA-DECIMAL TO DECIMAL\*\*\n");

Y:

printf("\nEnter the Number in Hexa-Decimal form: ");

scanf("%s",&hex);

//check

for(i=strlen(hex)-1;i>=0;i--)

{

if(hex[i]>'f' && hex[i]<='z' || hex[i]>'F'&& hex[i]<='Z')

{

printf("\nYou have to Enter Hexa-Decimal Number.\n");

printf("'%c' IS NOT Hexa-Decimal Number.\n",hex[i]);

goto Y;

}

}

Hex\_to\_Dec(hex); break;

case 12:

printf("\n\*\*HEXA-DECIMAL TO OCTAL\*\*\n");

Z:

printf("\nEnter the Number in Hexa-Decimal form: ");

scanf("%s",&hex);

//check

for(i=strlen(hex)-1;i>=0;i--)

{

if(hex[i]>'f' && hex[i]<='z' || hex[i]>'F'&& hex[i]<='Z')

{

printf("\nYou have to Enter Hexa-Decimal Number.\n");

printf("'%c' IS NOT Hexa-Decimal Number.\n",hex[i]);

goto Z;

}

}

Hex\_to\_Oct(hex); break;

default:

printf("\n\*\*INVALID NUMBER\*\*\n");

break;

}

printf("\n\nDO YOU WANT TO CONTINUE = (1/0) :\n");

scanf("%d",&num);

}

space = 3+35;

for( i=1;i<=3;i++)

{

for( j=1;j<=space;j++)

{

printf(" ");

}

space--;

for( j=1;j<=2\*i-1;j++)

{

printf("\*");

}

printf("\n");

}

space = 37;

for( i=1;i<=3;i++)

{

for( j=1;j<=space;j++)

{

printf(" ");

}

space++;

for( j=1;j<=2\*(3-i)-1;j++)

{

printf("\*");

}

printf("\n");

}

printf("\t\t BROUGHT TO YOU BY code-projects.org \n\t\t Log On now For More Free Projects");

}

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);

}

long int Bin\_to\_Oct(long int bin)

{

int i=0,rem,sum=0,remain[100],len=0;

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]);

}

}

long int Bin\_to\_Hex(long int bin)

{

int rem,i=0,sum=0,remain[100],len=0;

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]);

}

}

}

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]);

}

}

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]);

}

}

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]);

}

}

}

long int Oct\_to\_Bin(long int oct)

{

int rem[50],len=0,decimal=0,i=0,num,ans;

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]);

}

}

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);

}

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]);

}

}

}

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]);

}

}

}

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

//(a[i]>=0 || a[i]<=9)

{

num=hex[i]-48;

}

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

power++;

}

printf("\nEquivalent Decimal Number : %d",decimal);

}

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

//(a[i]>=0 || a[i]<=9)

{

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--)

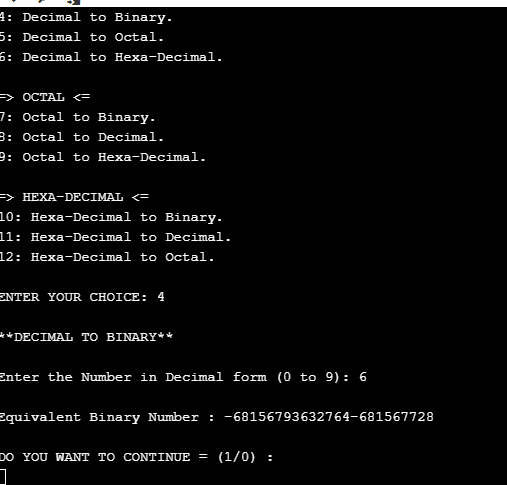
{

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

}

}

**Output:-**

****