1.

#include <stdio.h>

int main() {

long int n;

int count = 0;

printf("Enter an integer: ");

scanf("%ld", &n);

while (n != 0) {

n =n/10;

count++;

}

printf("Number of digits: %d", count);

return 0;

}

2.

#include <stdio.h>

#include <math.h>

int main {

int Number, FirstDigit, Digitcount, LastDigit, a, b, SwapNum,temp;

printf("Enter any number:");

scanf("%d",&Number);

Digitcount = log10(Number);

FirstDigit = Number / pow(10, Digitcount);

LastDigit = Number % 10;

a = FirstDigit \* (pow(10, Digitcount));

b = Number % a;

Number = b / 10;

SwapNum = LastDigit \* (pow(10, Digitcount)) + (Number \* 10 + FirstDigit);

printf(" \n The Number after Swapping First Digit and Last Digit = %d", SwapNum);

return 0;

}

3.

#include <stdio.h>

int main()

{

long int num, n;

int i, lastDigit;

int freq[10];

printf("Enter any number: ");

scanf("%ld", &num);

for(i=0; i<10; i++)

{

freq[i] = 0;

}

n = num;

while(n != 0)

{

lastDigit = n % 10;

n /= 10;

freq[lastDigit]++;

}

printf("Frequency of each digit in %ld is: \n", num);

for(i=0; i<10; i++)

{

printf("Frequency of %d = %d\n", i, freq[i]);

}

return 0; }

4.

#include <stdio.h>

int main()

{

long int n, num = 0;

printf("Enter any number: ");

scanf("%ld", &n);

while(n != 0)

{

num = (num \* 10) + (n % 10);

n /= 10;

}

while(num != 0)

{

switch(num % 10)

{

case 0:

printf("Zero ");

break;

case 1:

printf("One ");

break;

case 2:

printf("Two ");

break;

case 3:

printf("Three ");

break;

case 4:

printf("Four ");

break;

case 5:

printf("Five ");

break;

case 6:

printf("Six ");

break;

case 7:

printf("Seven ");

break;

case 8:

printf("Eight ");

break;

case 9:

printf("Nine ");

break;

}

num = num / 10;

}

return 0;

}

5.

#include <stdio.h>

int main()

{

char c;

printf("Enter a character: ");

scanf("%c", &c);

printf("ASCII value of %c = %d", c, c);

return 0;

}

6.

#include<stdio.h>

#include<stdlib.h>

#define SIZE 8

int main()

{

int i, bit\_size;

char num[SIZE + 1], one\_comp[SIZE + 1];

printf("Enter the binary number:\n");

gets(num);

for(i = 0; i < SIZE; i++)

{

if(num[i] == '0'){

one\_comp[i] = '1';

}

else if(num[i] == '1') {

one\_comp[i] = '0';

}}

one\_comp[SIZE] = '\0';

printf("Ones' complement of binary number %s is %s\n",num, one\_comp);

return 0;

}

7.

#include <stdio.h>

#define SIZE 8

int main()

{

char binary[SIZE + 1], onesComp[SIZE + 1], twosComp[SIZE + 1];

int i, carry=1;

printf("Enter binary value: ");

gets(binary);

for(i=0; i<SIZE; i++)

{

if(binary[i] == '1')

{

onesComp[i] = '0';

}

else if(binary[i] == '0')

{

onesComp[i] = '1';

}}

onesComp[SIZE] = '\0';

for(i=SIZE-1; i>=0; i--)

{

if(onesComp[i] == '1' && carry == 1)

{

twosComp[i] = '0';}

else if(onesComp[i] == '0' && carry == 1)

{

twosComp[i] = '1';

carry = 0;}

else

{

twosComp[i] = onesComp[i];

}}

twosComp[SIZE] = '\0';

printf("Original binary = %s\n", binary);

printf("Ones complement = %s\n", onesComp);

printf("Twos complement = %s\n", twosComp);

return 0;

}

8.

#include <stdio.h>

int main()

{

long int binarynum, octalnum = 0, j = 1, remainder;

printf("Enter the binary number: ");

scanf("%ld", &binarynum);

while (binarynum != 0)

{

remainder = binarynum % 10;

octalnum = octalnum + remainder \* j;

j = j \* 2;

binarynum = binarynum / 10; }

printf("Equivalent octal value: %lo", octalnum);

return 0;

}

9.

#include <stdio.h>

int main()

{

int num, binary\_val, decimal\_val = 0, base = 1, rem;

printf("Enter a binary number :");

scanf("%d", &num);

binary\_val = num;

while (num > 0)

{

rem = num % 10;

decimal\_val = decimal\_val + rem \* base;

num = num / 10 ;

base = base \* 2;

}

printf("The Binary number is = %d \n", binary\_val);

printf("Its decimal equivalent is = %d \n", decimal\_val);

return 0;

}

10.

#include<stdio.h>

int main()

{

long int binary\_number, hexadecimal\_number = 0, i = 1, remainder;

printf("Enter any Binary Number: ");

scanf("%ld", &binary\_number);

while (binary\_number != 0)

{

remainder = binary\_number % 10;

hexadecimal\_number = hexadecimal\_number + remainder \* i;

i = i \* 2;

binary\_number = binary\_number / 10;

}

printf("Hexadecimal Number:%lX", hexadecimal\_number);

return 0;

}

11.

#include <stdio.h>

#include <math.h>

int main()

{

long int n1,temp,p=1,dec=0,i=1,j,d,binnum=0;

printf("Input an octal number:");

scanf("%ld",&n1);

temp=n1;

for (j=n1;j>0;j=j/10)

{

d = j % 10;

if(i==1)

p=p\*1;

else

p=p\*8;

dec=dec+(d\*p);

i++; }

i=1;

for(j=dec;j>0;j=j/2)

{

binnum=binnum+(dec % 2)\*i;

i=i\*10;

dec=dec/2; }

printf("\nThe Octal Number :%ld \nBinary number:%ld",temp,binnum);

return 0;

}

12.

include <stdio.h>

#include <math.h>

long octalToDecimal(int octalnum)

{

int decimalnum = 0, temp = 0;

while(octalnum != 0)

{

decimalnum = decimalnum + (octalnum%10) \* pow(8,temp);

temp++;

octalnum = octalnum / 10;}

return decimalnum;}

int main()

{

int octalnum;

printf("Enter an octal number: ");

scanf("%d", &octalnum);

printf("Decimal number:%ld", octalToDecimal(octalnum));

return 0;

}

13.

#include<stdio.h>

#include<string.h>

int main()

{

int octnum, rev=0, rem, count=0, hex=0, mul=1, i=0, k=0;

char binnum[40] = "", hexnum[40];

printf("Enter any Octal Number: ");

scanf("%d", &octnum);

while(octnum!=0)

{

rem = octnum%10;

if(rem>7)

{

count++;

break;

}

rev = (rev\*10) + rem;

octnum = octnum/10;

}

if(count==0)

{

octnum = rev;

while(octnum!=0)

{

rem = octnum%10;

switch(rem)

{

case 0: strcat(binnum, "000");

break;

case 1: strcat(binnum, "001");

break;

case 2: strcat(binnum, "010");

break;

case 3: strcat(binnum, "011");

break;

case 4: strcat(binnum, "100");

break;

case 5: strcat(binnum, "101");

break;

case 6: strcat(binnum, "110");

break;

case 7: strcat(binnum, "111");

break;

}

octnum = octnum/10;

}

while(binnum[k]!='\0')

k++;

count=1;

k--;

while(k>=0)

{

if(binnum[k]=='0')

rem = 0;

else

rem = 1;

hex = hex + (rem\*mul);

if(count%4==0)

{

if(hex<10)

hexnum[i] = hex+48;

else

hexnum[i] = hex+55;

mul = 1;

hex = 0;

count = 1;

i++;

}

else

{

mul = mul\*2;

count++;

}

k--;

}

if(count!=1)

hexnum[i] = hex+48;

if(count==1)

i--;

printf("\nEquivalent Hexadecimal Value = ");

count = 0;

for(i=i; i>=0; i--)

{

if(hexnum[i]=='0' && count==0)

{

count++;

continue;

}

else

printf("%c", hexnum[i]);

}

}

else

printf("\nInvalid Octal Digit %d", rem);

return 0;

}

14.

#include<stdio.h>

#include<stdlib.h>

int main(){

int a[10],n,i;

printf("Enter the number to convert: ");

scanf("%d",&n);

for(i=0;n>0;i++)

{

a[i]=n%2;

n=n/2;

}

printf("\nBinary of Given Number is=");

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

{

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

}

return 0;

}

15.

#include <stdio.h>

#include <math.h>

int convertDecimalToOctal(int decimalNumber);

int main()

{

int decimalNumber;

printf("Enter a decimal number: ");

scanf("%d", &decimalNumber);

printf("%d in decimal = %d in octal", decimalNumber, convertDecimalToOctal(decimalNumber));

return 0;

}

int convertDecimalToOctal(int decimalNumber)

{

int octalNumber = 0, i = 1;

while (decimalNumber != 0)

{

octalNumber += (decimalNumber % 8) \* i;

decimalNumber /= 8;

i \*= 10;

}

return octalNumber;

}

16.

#include <stdio.h>

int main()

{

long decimalnum, quotient, remainder;

int i, j = 0;

char hexadecimalnum[100];

printf("Enter decimal number: ");

scanf("%ld", &decimalnum);

quotient = decimalnum;

while (quotient != 0)

{

remainder = quotient % 16;

if (remainder < 10)

hexadecimalnum[j++] = 48 + remainder;

else

hexadecimalnum[j++] = 55 + remainder;

quotient = quotient / 16;

}

for (i = j; i >= 0; i--)

printf("%c", hexadecimalnum[i]);

return 0;

}

17.

#include <stdio.h>

#define MAX 1000

int main()

{

char binarynum[MAX], hexa[MAX];

long int i = 0;

printf("Enter the value for hexadecimal ");

scanf("%s", hexa);

printf("\n Equivalent binary value: ");

while (hexa[i])

{

switch (hexa[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':

printf("1010"); break;

case 'B':

printf("1011"); break;

case 'C':

printf("1100"); break;

case 'D':

printf("1101"); break;

case 'E':

printf("1110"); break;

case 'F':

printf("1111"); break;

case 'a':

printf("1010"); break;

case 'b':

printf("1011"); break;

case 'c':

printf("1100"); break;

case 'd':

printf("1101"); break;

case 'e':

printf("1110"); break;

case 'f':

printf("1111"); break;

default:

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

return 0;

}

i++;

}

return 0;

}

18.

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

int main()

{

int i=0, chk=0, len, rem, binDigit, octalDigit;

long long binaryInt=0, temp=1, octalNum, binaryNum;

char hexdec[11], binnum[40]="";

printf("Enter Hexadecimal Number: ");

gets(hexdec);

while(hexdec[i])

{

switch(hexdec[i])

{

case '0':

strcat(binnum, "0000");

break;

case '1':

strcat(binnum, "0001");

break;

case '2':

strcat(binnum, "0010");

break;

case '3':

strcat(binnum, "0011");

break;

case '4':

strcat(binnum, "0100");

break;

case '5':

strcat(binnum, "0101");

break;

case '6':

strcat(binnum, "0110");

break;

case '7':

strcat(binnum, "0111");

break;

case '8':

strcat(binnum, "1000");

break;

case '9':

strcat(binnum, "1001");

break;

case 'A':

strcat(binnum, "1010");

break;

case 'a':

strcat(binnum, "1010");

break;

case 'B':

strcat(binnum, "1011");

break;

case 'b':

strcat(binnum, "1011");

break;

case 'C':

strcat(binnum, "1100");

break;

case 'c':

strcat(binnum, "1100");

break;

case 'D':

strcat(binnum, "1101");

break;

case 'd':

strcat(binnum, "1101");

break;

case 'E':

strcat(binnum, "1110");

break;

case 'e':

strcat(binnum, "1110");

break;

case 'F':

strcat(binnum, "1111");

break;

case 'f':

strcat(binnum, "1111");

break;

default:

chk = 1;

break;

}

i++;

}

if(chk==0)

{

len = strlen(binnum);

while(len!=0)

{

if(binnum[len-1]=='0')

binDigit=0;

else

binDigit=1;

binaryInt = binaryInt + (binDigit\*temp);

temp = temp\*10;

len--;

}

binaryNum = binaryInt;

octalNum = 0;

temp = 1;

while(binaryNum>0)

{

rem = binaryNum%1000;

switch(rem)

{

case 0:

octalDigit = 0;

break;

case 1:

octalDigit = 1;

break;

case 10:

octalDigit = 2;

break;

case 11:

octalDigit = 3;

break;

case 100:

octalDigit = 4;

break;

case 101:

octalDigit = 5;

break;

case 110:

octalDigit = 6;

break;

case 111:

octalDigit = 7;

break;

}

octalNum = (octalDigit\*temp) + octalNum;

binaryNum = binaryNum/1000;

temp = temp\*10;

}

printf("\nEquivalent Octal Value = %lld", octalNum);

}

else

printf("\nInvalid Hexadecimal Character/Digit");

getch();

return 0;

}

19.

#include<stdio.h>

#include<conio.h>

#include<math.h>

int main()

{

int decnum=0, rem, i=0, len=0;

char hexnum[20];

printf("Enter any Hexadecimal Number: ");

scanf("%s", hexnum);

while(hexnum[i]!='\0')

{

len++;

i++;

}

len--;

i=0;

while(len>=0)

{

rem = hexnum[len];

if(rem>=48 && rem<=57)

rem = rem-48;

else if(rem>=65 && rem<=70)

rem = rem-55;

else if(rem>=97 && rem<=102)

rem = rem-87;

else

{

printf("\nYou've entered an invalid Hexadecimal digit");

getch();

return 0;

}

decnum = decnum + (rem\*pow(16, i));

len--;

i++;

}

printf("\nEquivalent Decimal Value = %d", decnum);

getch();

return 0;

}