**CA3001 – Programming and Data Structures using C**

**Assignment 9 - 01.02.2021**

1. Functions without arguments and without return type

• check whether the year is Leap year

• convert binary to hexadecimal

• count number of digits in a number

Ans – C Program & Output:

* check whether the year is Leap year

#include <stdio.h>  
void leapyear();  
int main()

{  
leapyear();  
return 0;  
}  
void leapyear()  
{  
int y;  
printf("please enter a year to check leap year or not:\n");  
scanf("%d",&y);  
   if(y%400==0 || y%4==0&&y%100!=0)  
   printf("%d is a leap year",y);  
   else  
 printf("%d is not a leap year",y);  
 }



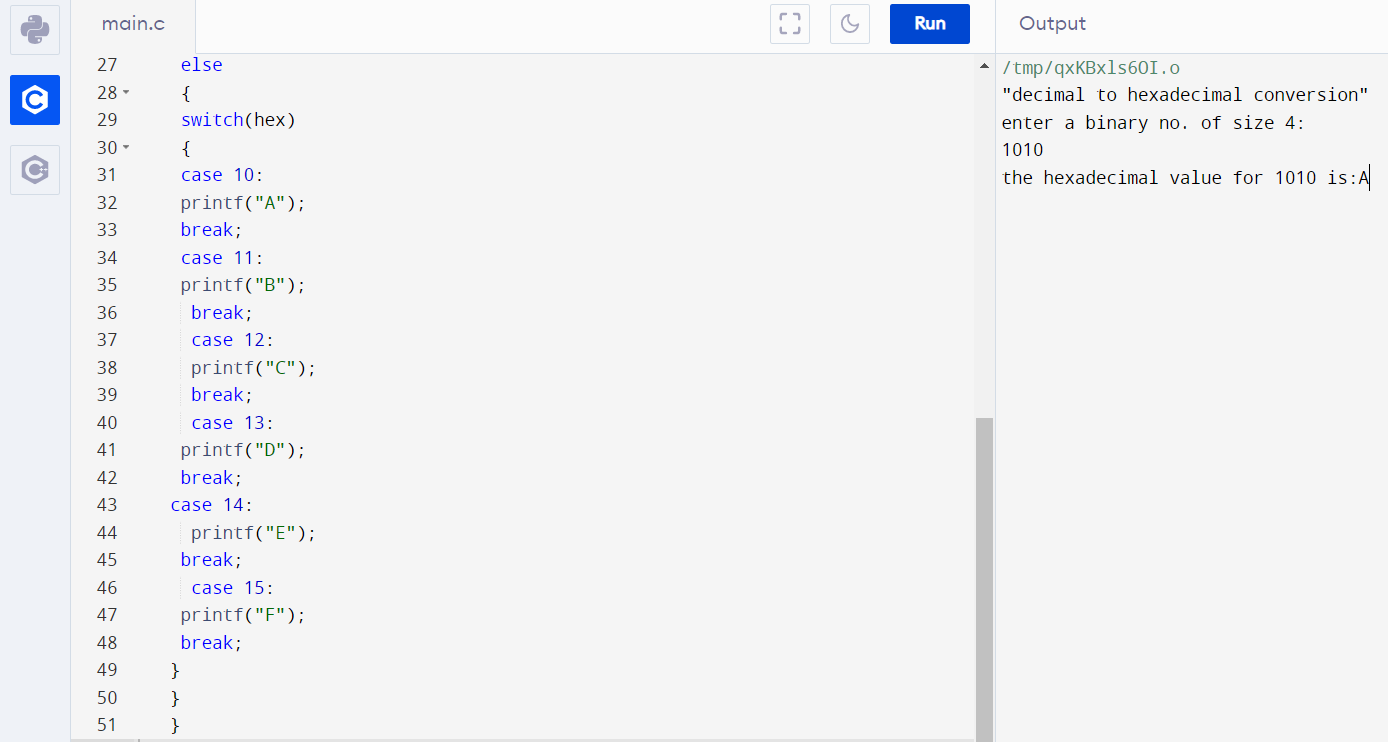
• convert binary to hexadecimal

#include <stdio.h>  
#include<math.h>  
void conversion ();  
int main()

{  
printf("\"decimal to hexadecimal conversion\"\n");  
conversion();  
return 0;  
}  
void conversion()

{  
 int b,a;  
 int hex=0,p=0;  
 printf("enter a binary no. of size 4:\n");  
 scanf("%d",&b);  
 printf("the hexadecimal value for %d is:",b);  
 while(b!=0)  
 {  
   a=b%10\*pow(2,p);  
  hex=hex+a;  
   b=b/10;  
   p++;  
    }  
    if(hex<=9)

{  
    printf("%d",hex);}  
    else

{  
    switch(hex)  
    {  
    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;  
   }  
   }  
 }  


• count number of digits in a number

#include <stdio.h>  
void count();  
int main()

{  
printf("count the number of digits in a number\n");  
count();  
return 0;  
}  
void count()  
{  
int num,i=0;  
printf("enter a number:\n");  
scanf("%d",&num);  
printf("the number of digits in %d is:",num);  
while(num!=0)  
{  
num=num/10;   
i++;  
}  
printf("%d",i);  
}



**b.** Functions without arguments and with return type

• check Armstrong number or not

• to evaluate the following using loops x + x^3 / 3! + x^5 / 5! +... upto 5 terms

• Convert temperature Fahrenheit to Celsius

Ans – C Program & Output:

• check Armstrong number or not

#include <stdio.h>

int arms();

int main()

{

int result;

result=arms();

if(result==1)

printf(" armstrong number");

else

printf(" not armstrong number");

return 0;

}

int arms()

{

int n,sum=0;

int num;

printf("enter a number to check armstrong or not:\n");

scanf("%d",&num);

printf("%d number is",num);

while(num!=0)

{

n=num % 10;

num=num/10;

sum=sum+(n\*n\*n);

}

if(sum==num)

{

return 0;

}

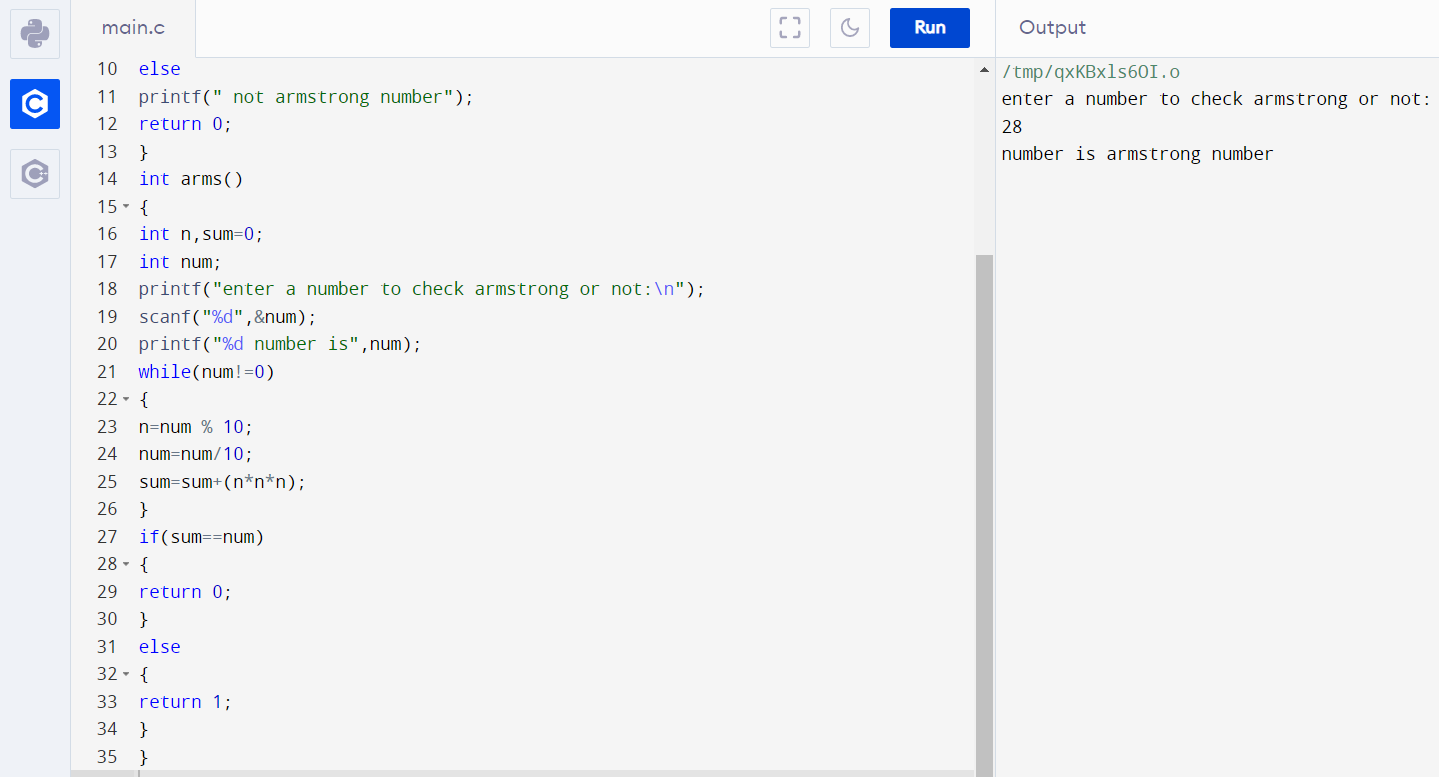
else

{

return 1;

}

}



• to evaluate the following using loops x + x^3 / 3! + x^5 / 5! +... upto 5 terms

#include <stdio.h>

#include <math.h>

float cal();

int main()

{

float res;

res=cal();

printf("is %.3f",res);

}

float cal()

{

int i=1,j=1,x,t;

float fac=1,sum=1;

printf("enter the value for x:\n");

scanf("%d",&x);

printf("the sum of ");

while(i<10)

{

while(j<i){

fac\*=j;

j++;

}

t=pow(x,i);

sum+=t/fac;

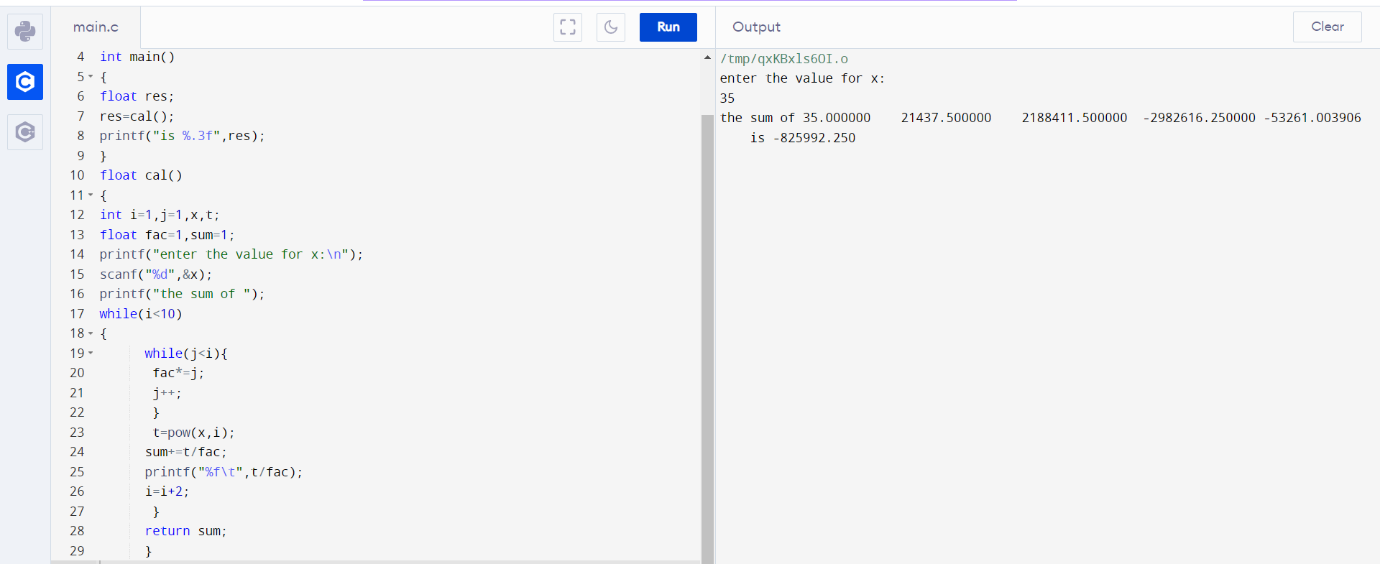
printf("%f\t",t/fac);

i=i+2;

}

return sum;

}



• Convert temperature Fahrenheit to Celsius

#include <stdio.h>

float temp();

int main()

{

printf("the value in fahrenheit is:%.3f",temp());

return 0;

}

float temp()

{

float c,f;

printf("enter value in centigrade for conversion to fahrenheit:\n");

scanf("%f",&c);

f=9/5\*c+32;

return f;

}



**c.** Functions with arguments and without return type

• check prime number or not

• find all roots of the quadratic equation

• find ASCII number to character and character to ASCII number

Ans – C Program & Output:

• check prime number or not

#include <stdio.h>

void prime(int);

int main()

{

int n;

printf("enter a number to determine prime or not:\n");

scanf("%d",&n);

prime(n);

}

void prime(int n)

{

int p,i=2;

do

{

if(n%i==0||n==1)

{

p=0;

break;

}

i++;

}

while(i<n/2);

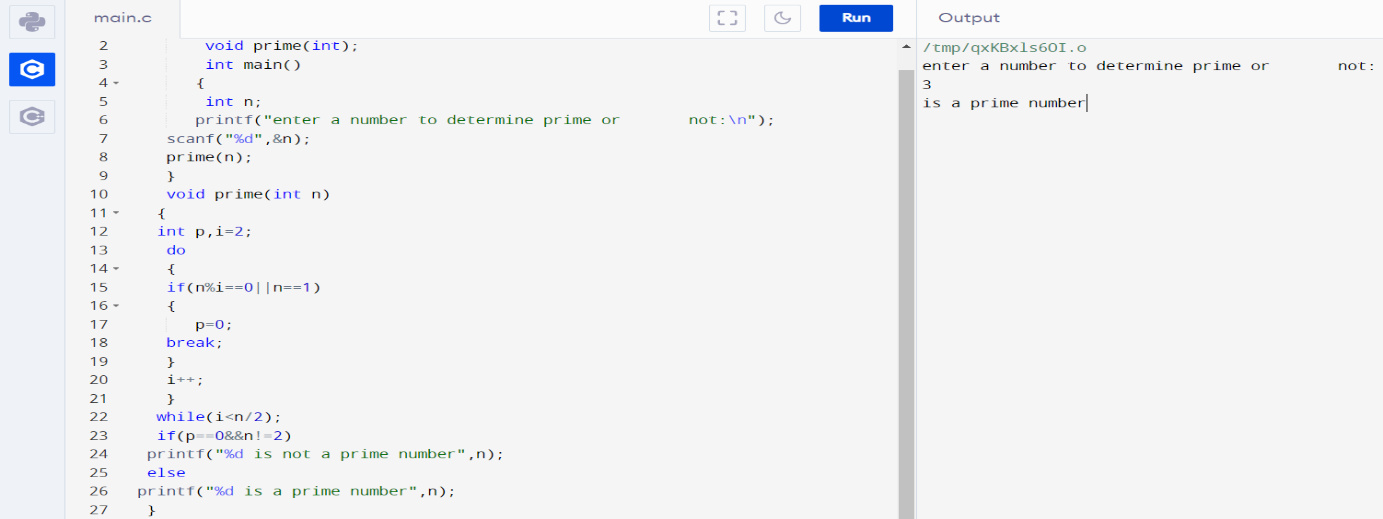
if(p==0&&n!=2)

printf("%d is not a prime number",n);

else

printf("%d is a prime number",n);

}



• find all roots of the quadratic equation

#include<stdio.h>

#include<math.h>

void quad(int,int,int);

int main()

{

int a,b,c;

printf("enter the value of a,b & c:\n ");

scanf("%d%d%d",&a,&b,&c);

quad(a,b,c);

return 0;

}

void quad(int a, int b, int c)

{

int d;

float r1,r2;

d=b\*b-4\*a\*c;

if(d==0)

{

printf("Both roots are equal.\n");

r1=r2=-b/(2.0\*a);

printf("First Root Root1: %f\n",r1);

printf("Second Root Root2: %f\n",r2);

}

else if(d>0)

{

printf("Both roots are real and different.\n");

r1=(-b+sqrt(d))/(2\*a);

r2=(-b-sqrt(d))/(2\*a);

printf("First Root Root1: %f\n",r1);

printf("Second Root root2: %f\n",r2);

}

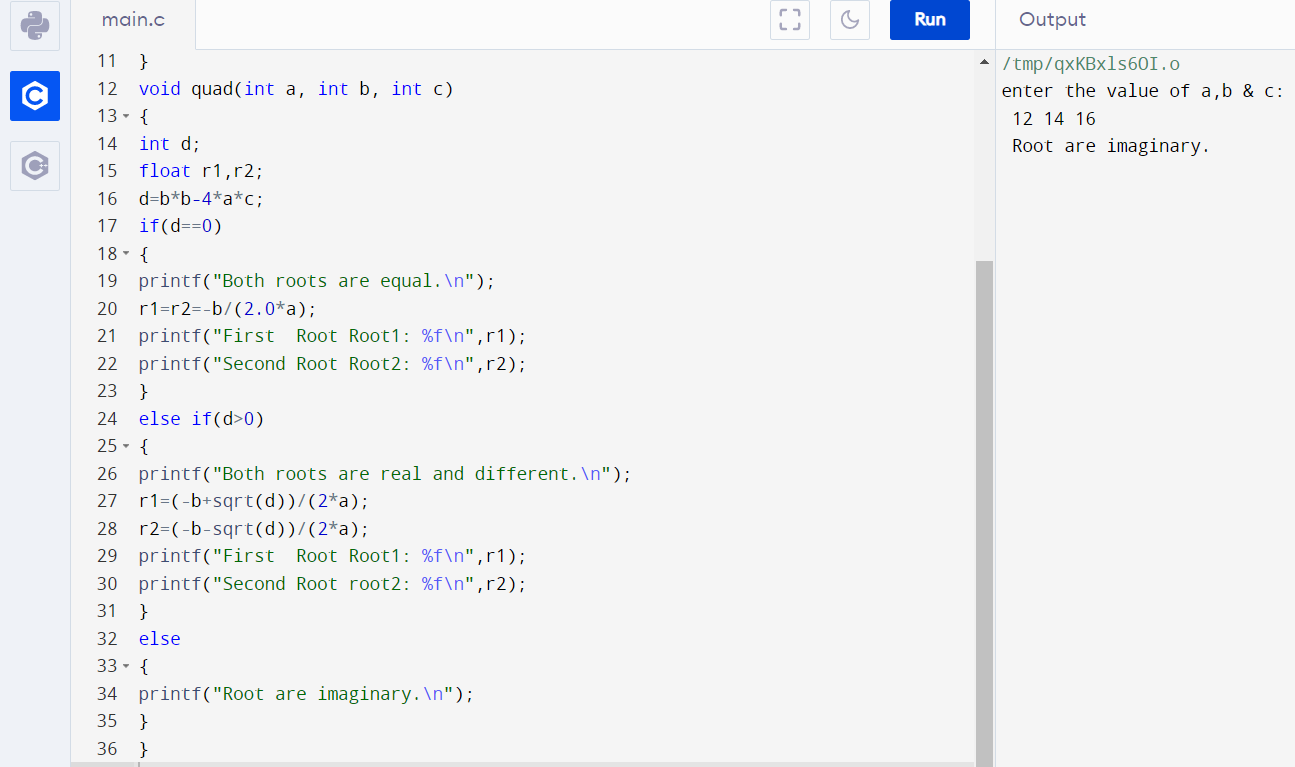
else

{

printf("Root are imaginary.\n");

}

}



• find ASCII number to character and character to ASCII number

#include <stdio.h>

void conversion(int,char);

int main()

{

int a;

char b;

printf("enter a character:\n");

scanf("%c",&b);

printf("enter a number:\n");

scanf("%d",&a);

conversion(a,b);

return 0;

}

void conversion(int a,char b){

char c;

printf("ASCII value of %c is %d\n", b, b);

c = a;

printf("the character assigned to ASCII value %d is %c", c,c);

}



**d.** Functions with arguments and with return type

• check perfect or abundant or deficient number

• calculate factorial of a number

• count number of digits in a number

• check perfect or abundant or deficient number #include <stdio.h>

int perfect(int);

int main()

{

int n,res;

printf("enter a number to check perfect or not : ");

scanf("%d",&n);

res=perfect(n);

if(res==n)

printf("the number is perfect.");

else if(res>n)

printf("the number is abundant.");

else

printf("the number is deficient.");

return 0;

}

int perfect(int n)

{

int i=1,sum=0;

printf("The factors are : ");

while(i<n)

{

if(n%i==0)

{

sum=sum+i;

printf("%d\t",i);

}

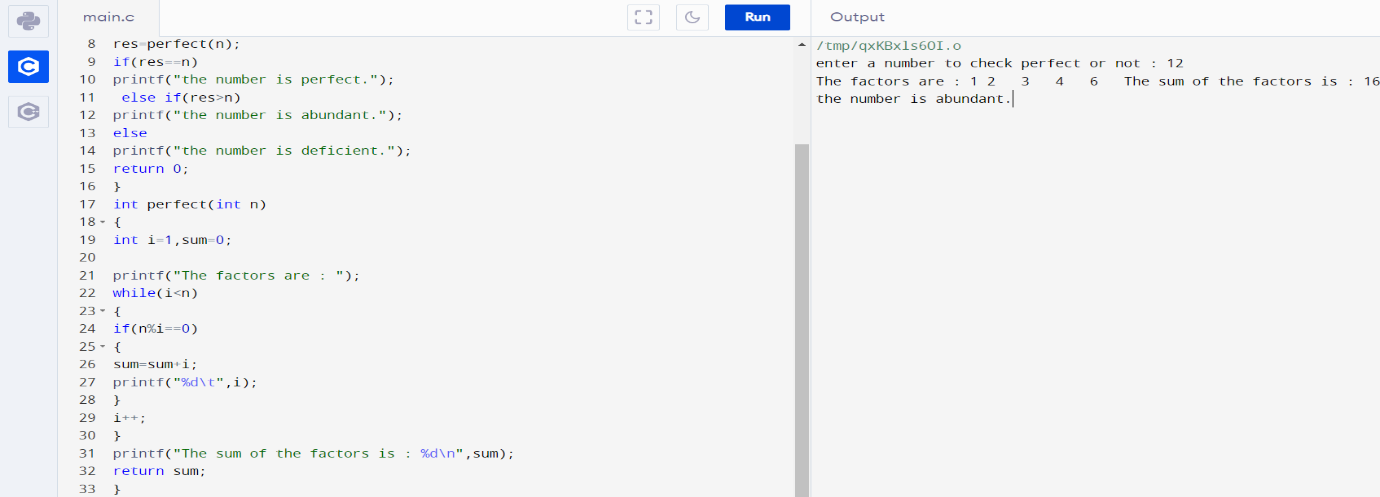
i++;

}

printf("The sum of the factors is : %d\n",sum);

return sum;

}



• calculate factorial of a number

#include<stdio.h>

int fac(int);

int main()

{

int n,res;

printf("enter a number:\n");

scanf("%d",&n);

res=fac(n);

printf("the factorial of %d is: %d",n,res);

return 0;

}

int fact(int n)

{

int f=1;

while(n!=0)

{

f=f\*n;

n—

}

return f;

}

• count number of digits in a number

#include <stdio.h>

int count(int);

int main()

{

int num,res;

printf("count the number of digits in a number\n");

printf("enter a number:\n");

scanf("%d",&num);

res=count(num);

printf("the number of digits in %d is %d",num,res);

return 0;

}

int count(int num)

{

int i=0;

while(num!=0)

{

num=num/10;

i++;

}

return i;

}



**e.** Function return Multiple values

* Largest and Smallest of five numbers
* Find Simple interest and compound interest
* simple calculator (add, sub, mul, div, mod)

Ans – C Program & Output:

* Largest and Smallest of five numbers

#include <stdio.h>

int max(int num1, int num2);

int min(int num1, int num2);

int main()

{

int num1, num2, maximum, minimum;

printf("Enter any two numbers: ");

scanf("%d%d", &num1, &num2);

maximum = max(num1, num2);

minimum = min(num1, num2);

printf("\nMaximum = %d\n", maximum);

printf("Minimum = %d", minimum);

return 0;

}

int max(int num1, int num2)

{

return (num1 > num2 ) ? num1 : num2;

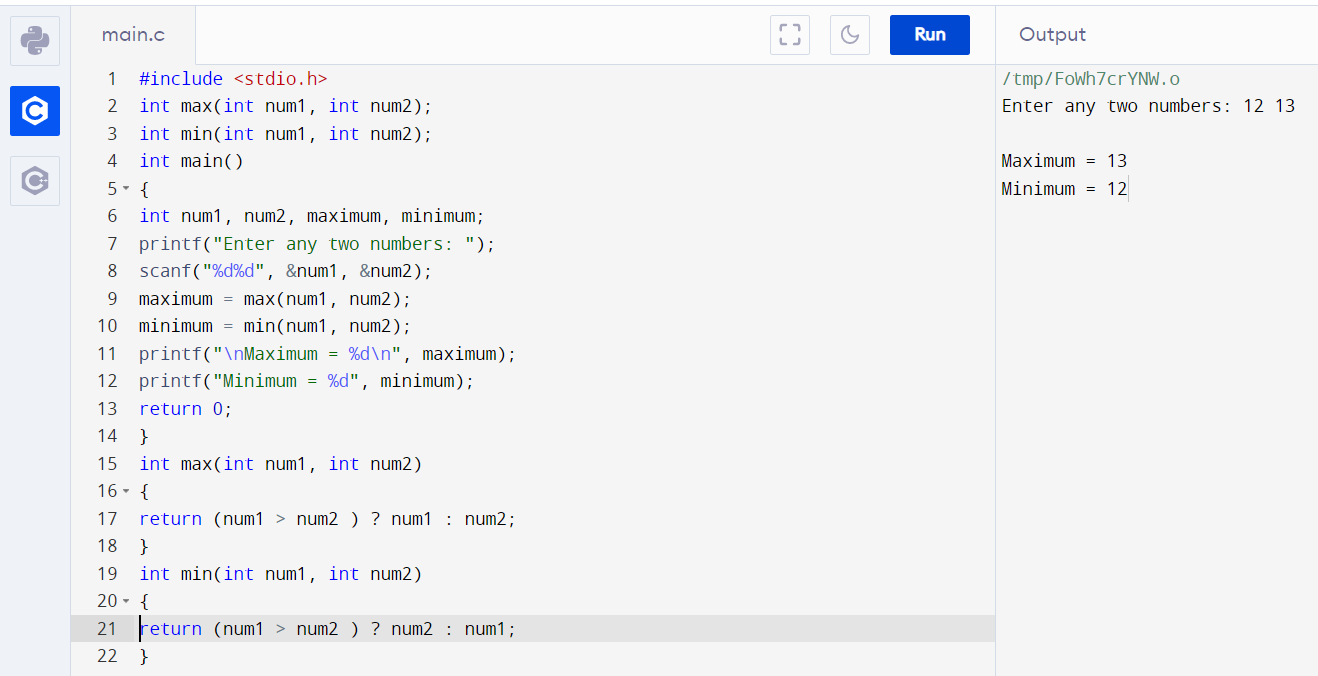
}

int min(int num1, int num2)

{

return (num1 > num2 ) ? num2 : num1;

}



* simple calculator (add, sub, mul, div, mod)

#include <stdio.h>

float add(float num1, float num2);

float sub(float num1, float num2);

float mult(float num1, float num2);

float div(float num1, float num2);

int main()

{

char op;

float num1, num2, result=0.0f;

scanf("%f %c %f", &num1, &op, &num2);

switch(op)

{

case '+':

result = add(num1, num2);

break;

case '-':

result = sub(num1, num2);

break;

case '\*':

result = mult(num1, num2);

break;

case '/':

result = div(num1, num2);

break;

default:

printf("Invalid operator");

}

printf("%.2f %c %.2f = %.2f", num1, op, num2, result);

return 0;

}

float add(float num1, float num2)

{

return num1 + num2;

}

float sub(float num1, float num2)

{

return num1 - num2;

}

float mult(float num1, float num2)

{

return num1 \* num2;

}

float div(float num1, float num2)

{

return num1 / num2;

}



**f.** Nesting of Functions

* Print the sum of series 1 + 1/2 + 1/3 + 1/4 + ... + 1/N.
* Find GCD and LCM of numbers
* reverse a number

Ans – C Program & Output:

* Print the sum of series 1 + 1/2 + 1/3 + 1/4 + ... + 1/N.

#include <stdio.h>

double sum(int n)

{

double i, s = 0.0;

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

s = s + 1/i;

return s;

}

int main()

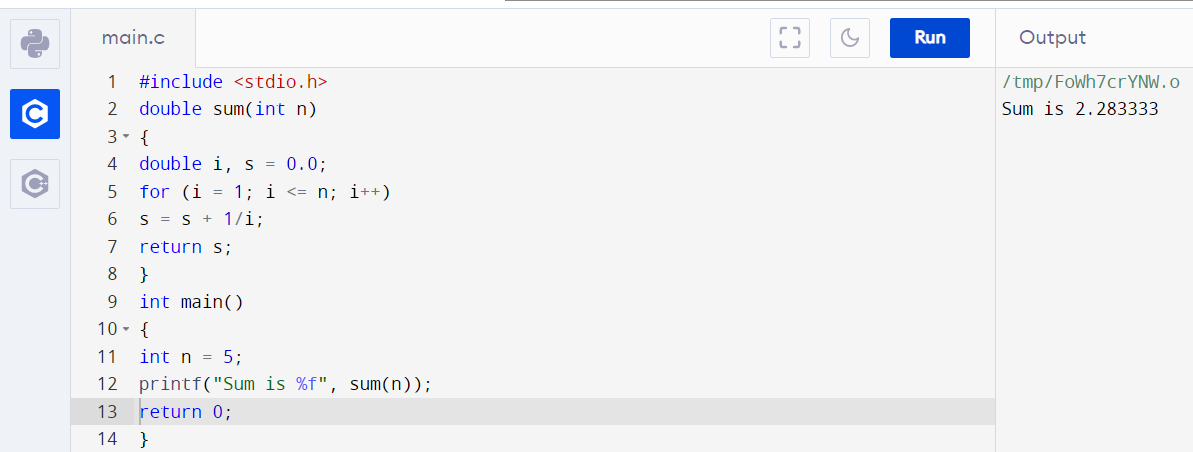
{

int n = 5;

printf("Sum is %f", sum(n));

return 0;

}



* Find GCD and LCM of numbers

#include <stdio.h>

long gcd(long, long);

int main()

{

long x, y, hcf, lcm;

printf("Enter two integers\n");

scanf("%ld%ld", &x, &y);

hcf = gcd(x, y);

lcm = (x\*y)/hcf;

printf("GCD of %ld and %ld = %ld\n", x, y, hcf);

printf("LCM OF %ld and %ld = %ld\n", x, y, lcm);

return 0;

}

long gcd(long x, long y) {

if (x == 0) {

return y;

}

while (y != 0) {

if (x > y)

x = x - y;

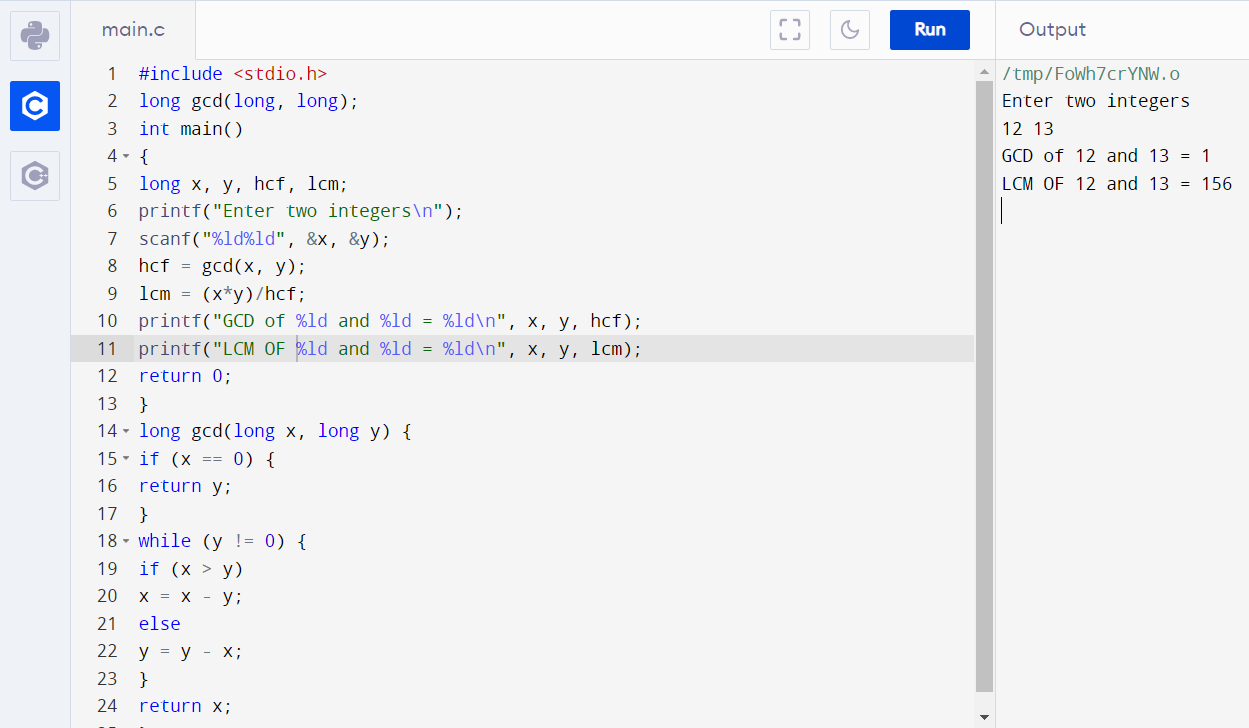
else

y = y - x;

}

return x;

}



* reverse a number

#include <stdio.h>

long reverse(long);

int main()

{

long n, r;

printf("Enter a number: \n");

scanf("%ld", &n);

r = reverse(n);

printf("%ld\n", r);

return 0;

}

long reverse(long n)

{

static long r = 0;

if (n == 0)

return 0;

r = r \* 10;

r = r + n % 10;

reverse(n/10);

return r;

}



**g.** Recursive Functions

* to Print Fibonacci Series
* to print even or odd numbers in given range
* to convert a decimal number to binary

Ans – C Program & Output :

* to Print Fibonacci Series

#include<stdio.h>

int Fibonacci(int);

int main()

{

int n, i = 0, c;

printf("Enter no. of terms of fibonacci series:\n");

scanf("%d",&n);

printf("Fibonacci series\n");

for ( c = 1 ; c <= n ; c++ )

{

printf("%d\n", Fibonacci(i));

i++;

}

return 0;

}

int Fibonacci(int n)

{

if ( n == 0 )

return 0;

else if ( n == 1 )

return 1;

else

return ( Fibonacci(n-1) + Fibonacci(n-2) );

}



* to print even or odd numbers in given range

#include <stdio.h>

void EvenAndOdd(int Val, int n);

int main()

{

int n;

printf(" Input the range to print starting from 1 : ");

scanf("%d", &n);

printf("\n All even numbers from 1 to %d are : ", n);

EvenAndOdd(2, n);

printf("\n All odd numbers from 1 to %d are : ", n);

EvenAndOdd(1, n);

printf("\n");

return 0;

}

void EvenAndOdd(int Val, int n)

{

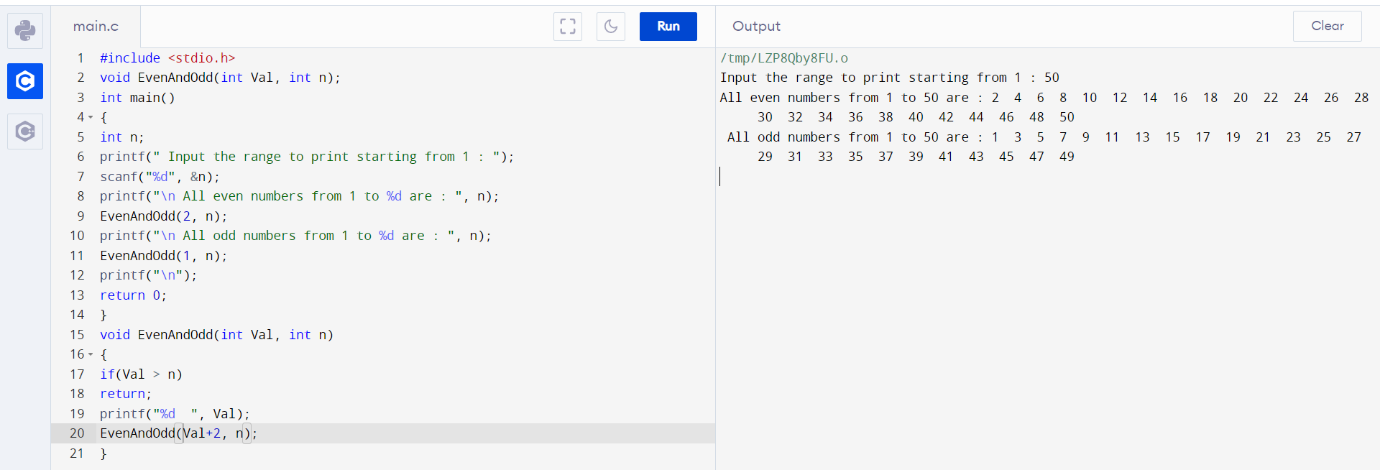
if(Val > n)

return;

printf("%d ", Val);

EvenAndOdd(Val+2, n);

}



* to convert a decimal number to binary

#include <stdio.h>

int decimalToBinary(int n)

{

if (n == 0)

{

return 0;

}

else

{

return (n % 2 + 10 \*

decimalToBinary(n / 2));

}

}

int main()

{

int num;

printf("Enter decimal number: ");

scanf("%d", &num);

printf ("Binary form of given decimal number is: %d",decimalToBinary(num));

return 0;

}

