**Question 1:**

**Code**

#include<stdio.h>

double factorial(int num);

void main (void){

int num;

double fact;

num=0 ;

printf("Enter the number to find its factorial: \n");

scanf("%d",&num);

fact = factorial(num);

printf("%lf",fact);

}

double factorial(int num){

if (num>=1)

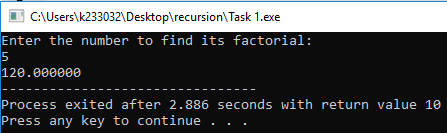
return num\* factorial(num-1);

else

return 1;

}

**Output**



**Question 2:**

**Code**

#include<stdio.h>

// without using multiplication operator and loops. Use recursion and addition only.

int multiplication(int num1 , int num2);

void main (void){

int num1=0, num2=0, multiply;

printf("Enter the two numbers: \n");

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

if (num1<0 || num2<0){

printf("You have eneterd negative numbers \n");

}

else if (num1>0 && num2>0){

printf("\nThe multiplication of the input numbers is %d",multiplication(num1,num2));

}

}

int multiplication(int num1, int num2){

if (num2>0){

return num1+multiplication(num1,num2-1);

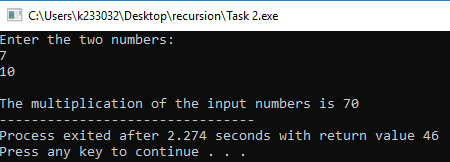
}

else

return 0;

}

**Output**



**Question 3:**

**Code**

#include<stdio.h>

int loop(int a, int b);

int GCD(int a, int b);

void main (void){

int a,b,gcd;

a=b=gcd=0;

printf("Enter the first number = \n");

scanf("%d",&a);

printf("Enter the second number = \n");

scanf("%d",&b);

gcd=loop(a,b);

printf("The gcd of the numbers %d and %d is %d\n",a,b,gcd);

printf("The GCD of the two numbers %d and %d is %d",a,b,GCD(a,b));

}

int loop(int a, int b){

int gcd=0;

for (int i=1; i<=a && i<=b; i++){

if (a%i==0 && b%i==0){

gcd=i;

}

}

return gcd;

}

int GCD(int a, int b){

if (b==0){

return a;

}

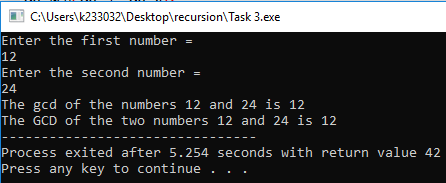
else{

return GCD(b, a % b);

}

}

**Output**



**Question 4:**

**Code**

#include<stdio.h>

int fibonacci(int num);

void main (void){

int i=0, number=0;

printf("Enter the number to print Fibonacci Series: \n");

scanf("%d",&number);

for (i=0; i<number; i++){

printf("%d, ",fibonacci(i));

}

}

int fibonacci(int n){

if (n==0){

return 0;

}

else if (n==1){

return 1;

}

else{

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

}

}

**Output**

