**Assignment - 13 A Job Ready Bootcamp in C++, DSA and IOT**

More on Recursion in C Language

**1. Write a recursive function to calculate sum of first N natural numbers**

#include<stdio.h>

int sumNatural(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("%d",sumNatural(x));

return 0;

}

int sumNatural(int n)

{

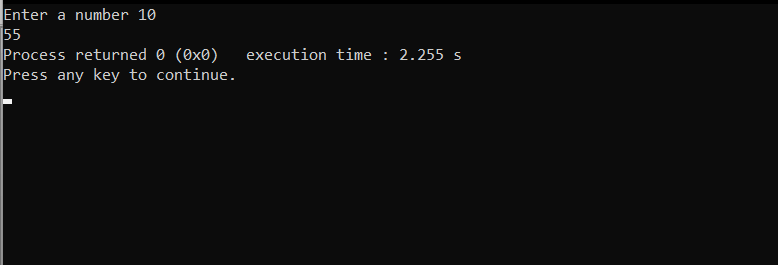
int sum=0;

if(n>0)

sum= n+sumNatural(n-1);

return sum;

}



**2. Write a recursive function to calculate sum of first N odd natural numbers**

#include<stdio.h>

int sumoddNatural(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("%d",sumoddNatural(x));

return 0;

}

int sumoddNatural(int n)

{

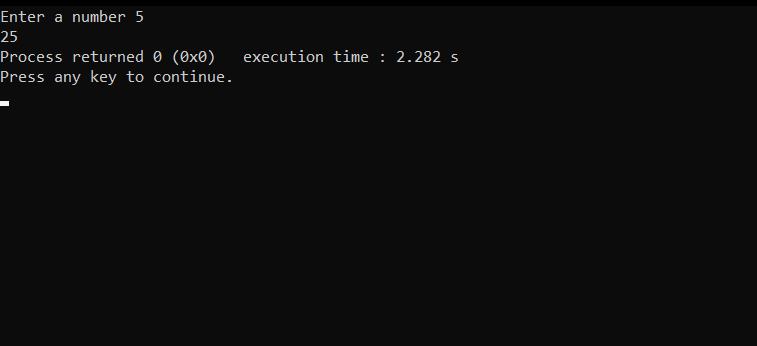
int sum=0;

if(n>0)

sum= (2\*n)-1 +sumoddNatural(n-1);

return sum;

}



**3. Write a recursive function to calculate sum of first N odd natural numbers**

#include<stdio.h>

int sumoddNatural(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("%d",sumoddNatural(x));

return 0;

}

int sumoddNatural(int n)

{

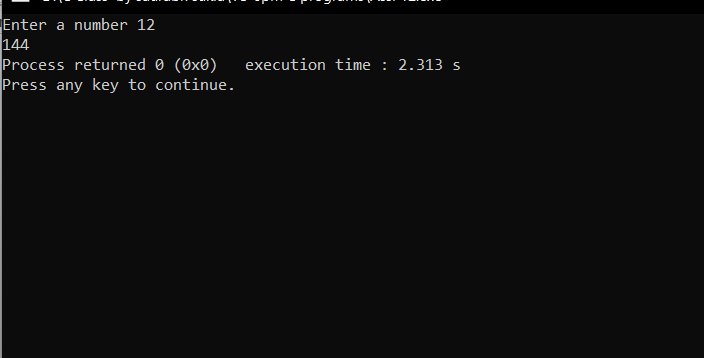
int sum=0;

if(n>0)

sum= (2\*n)-1 +sumoddNatural(n-1);

return sum;

}



**4. Write a recursive function to calculate sum of squares of first n natural numbers**

#include<stdio.h>

int squaresum(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("%d",squaresum(x));

return 0;

}

int squaresum(int n)

{

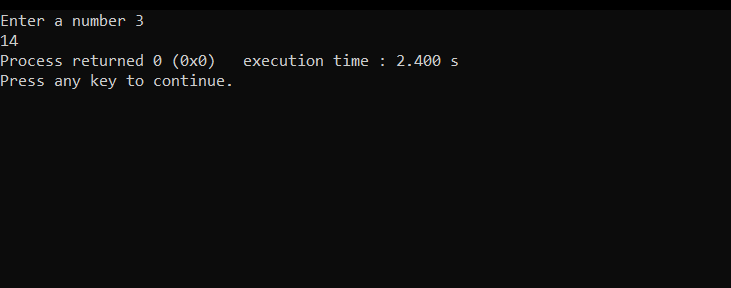
int sum=0;

if(n>0)

sum= (n\*n)+squaresum(n-1);

return sum;

}



**5. Write a recursive function to calculate sum of digits of a given number**

#include<stdio.h>

int sumdigit(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("%d", sumdigit(x));

return 0;

}

int sumdigit(int num)

{

int sum=0;

if(num>0)

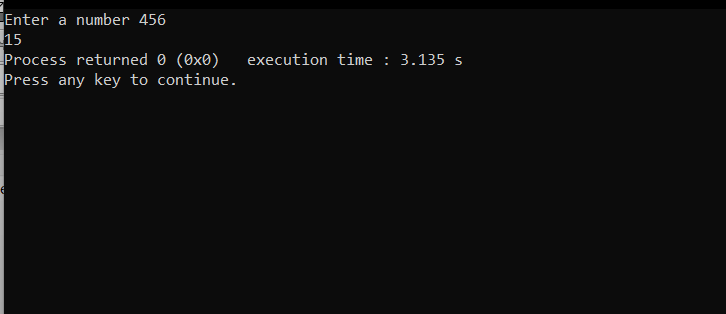
{

sum=sumdigit(num/10)+ num%10;

}

return sum;

}



**6. Write a recursive function to calculate factorial of a given number**

#include<stdio.h>

int factorial(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("%d",factorial(x));

return 0;

}

int factorial(int num)

{

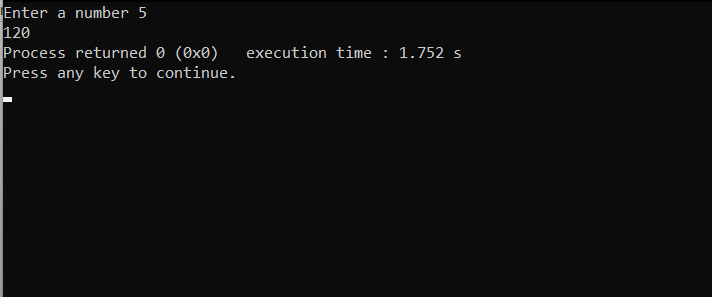
int fact=1;

if(num>0)

fact= num\*factorial(num-1);

return fact;

}



**7. Write a recursive function to calculate HCF of two numbers**

#include<stdio.h>

int hcf(int,int);

int main()

{

int x,y,c;

printf("Enter 1st number ");

scanf("%d",&x);

printf("Enter 2nd number ");

scanf("%d",&y);

if(x>y)

printf("%d", hcf(y,x%y));

else

c=x;

x=y;

y=c;

printf("%d", hcf(y,x%y));

return 0;

}

int hcf(int num1,int num2)

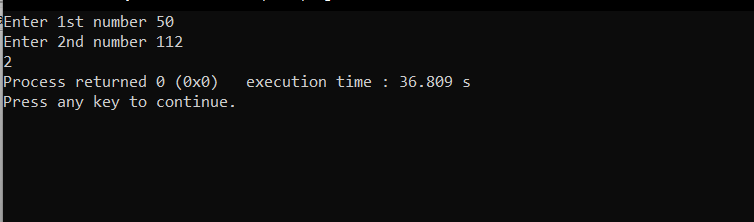
{

if(num2==0)

return num1;

hcf(num2,num1%num2);

}



**8. Write a recursive function to print first N terms of Fibonacci series**

#include<stdio.h>

void fib(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("1 ");

fib(x-1);

return 0;

}

void fib(int num)

{

static int prev=0,curr=1,next=1;

if(num>0)

{

next=prev+curr;

prev=curr;

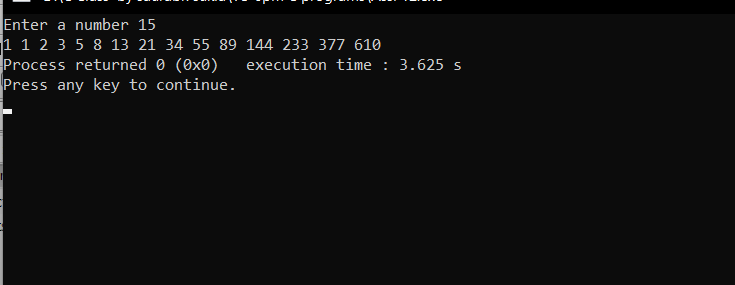
curr=next;

printf("%d ",next);

fib(num-1);

}

}



**9. Write a program in C to count the digits of a given number using recursion.**

#include<stdio.h>

int count(int);

int main()

{

int x;

printf("Enter a number ");

scanf("%d",&x);

printf("Number of digits in %d is %d",x,count(x));

return 0;

}

int count(int num)

{

static int c=0;

if(num>0)

{

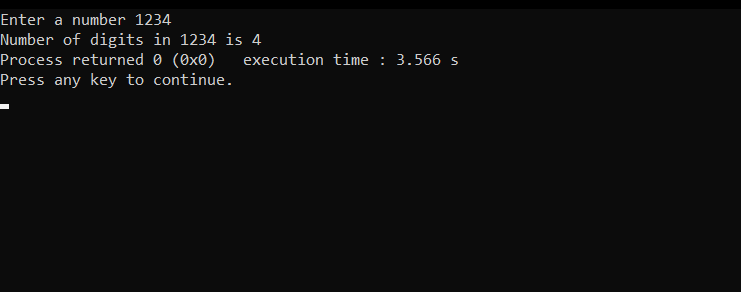
count(num/10);

c++;

}

return c;

}



**10. Write a program in C to calculate the power of any number using recursion.**

#include<stdio.h>

int calExponent(int,int);

int main()

{

int x,p;

printf("Enter a value ");

scanf("%d",&x);

printf("Enter its Exponent ");

scanf("%d",&p);

printf("%d",calExponent(x,p));

return 0;

}

int calExponent(int num,int expo)

{

static int result=1;

if(expo>0)

{

calExponent(num,expo-1);

result\*=num;

}

return result;

}

