Assignment -13

More on Recursion in C Language

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

Code

#include<stdio.h>

int sum\_of\_first\_n\_terms(int);

int main()

{     int n;

     printf("Enter number of terms :");

     scanf(" %d",&n);

     printf("Sum of odd terms: %d",sum\_of\_first\_n\_terms(n));

    return 0;

}

int sum\_of\_first\_n\_terms(int n)

  {

     int s;

    if(n==0)

    return 0;

   s= sum\_of\_first\_n\_terms(n-1) + n;

    //printf("%d",s);

    return s;

  }

Output

Enter number of terms :5

Sum of odd terms: 15

1. Write a recursive function to calculate sum of first N odd natural number

Code

\ #include<stdio.h>

int sum\_of\_first\_n\_odd\_terms(int);

int main()

{     int n;

     printf("Enter number of terms :");

     scanf(" %d",&n);

     printf("Sum of terms: %d",sum\_of\_first\_n\_odd\_terms(n));

    return 0;

}

int sum\_of\_first\_n\_odd\_terms(int n)

  {

     int s;

    if(n==0)

    return 0;

   s= sum\_of\_first\_n\_odd\_terms(n-1) + (2\*n-1);

    //printf("%d",s);

    return s;

  }

Output

Enter number of terms :3

Sum of terms: 9

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

Code

#include<stdio.h>

int sum\_of\_first\_n\_even\_terms(int);

int main()

{     int n;

     printf("Enter number of terms :");

     scanf(" %d",&n);

     printf("Sum of even terms: %d",sum\_of\_first\_n\_even\_terms(n));

    return 0;

}

int sum\_of\_first\_n\_even\_terms(int n)

  {

     int s;

    if(n==0)

    return 0;

   s= sum\_of\_first\_n\_even\_terms(n-1) + (2\*n);

    //printf("%d",s);

    return s;

  }

Output

Enter number of terms :2

Sum of even terms: 6

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

Code

#include<stdio.h>

int sum\_of\_first\_n\_square\_terms(int);

int main()

{     int n;

     printf("Enter number of terms :");

     scanf(" %d",&n);

     printf("Sum of square terms: %d",sum\_of\_first\_n\_square\_terms(n));

    return 0;

}

int sum\_of\_first\_n\_square\_terms(int n)

  {

     int s;

    if(n==0)

    return 0;

   s= sum\_of\_first\_n\_square\_terms(n-1) + (n\*n);

    //printf("%d",s);

    return s;

  }

Output

Enter number of terms :2

Sum of square terms: 5

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

Code

#include<stdio.h>

int sum\_of\_digit\_term(int);

int main()

{     int n;

     printf("Enter number of terms :");

     scanf(" %d",&n);

     printf("Sum of digit in a term: %d",sum\_of\_digit\_term(n));

    return 0;

}

int sum\_of\_digit\_term(int n)

  {

     int s;

    if(n==0)

    return 0;

   s= sum\_of\_digit\_term(n/10) + (n%10);

    //printf("%d",s);

    return s;

  }

Output

Enter number of terms :345

Sum of digit in a term: 12

1. Write a recursive function to calculate factorial of a given number

Code

#include<stdio.h>

int factorial\_of\_num(int);

int main()

{     int n;

     printf("Enter a number :");

     scanf(" %d",&n);

     printf("factorial of number: %d",factorial\_of\_num(n));

    return 0;

}

int factorial\_of\_num(int n)

  {

     int s;

    if(n==0)

    return 1;

   s= factorial\_of\_num(n-1)\*n;

    //printf("%d",s);

    return s;

  }

Output

Enter a number :5

factorial of number: 120

1. Write a recursive function to calculate HCF of two numbers

Code

#include<stdio.h>

int hcf\_of\_two\_num(int,int);

int main()

{     int n1,n2;

     printf("Enter a number N1 :");

     scanf(" %d",&n1);

     printf("Enter a number N2 :");

     scanf(" %d",&n2);

     printf("HCF of number: %d",hcf\_of\_two\_num(n1,n2));

    return 0;

}

int hcf\_of\_two\_num(int n1, int n2)

  {

     int s;

    if(n2==0)

    return n1;

   s= hcf\_of\_two\_num(n2, n1%n2);

    //printf("%d",s);

    return s;

  }

Output

Enter a number N1 :6

Enter a number N2 :4

factorial of number: 2

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

Code

#include<stdio.h>

int print\_first\_n\_fibinocci\_terms(int);

int main()

{     int n,i;

     printf("Enter number of terms :");

     scanf(" %d",&n);

    for(i=1;i<=n;i++)

    {

      printf(" %d ",print\_first\_n\_fibinocci\_terms(i));

    }

    return 0;

}

int print\_first\_n\_fibinocci\_terms(int n)

  {

     int s;

    if(n==1 || n==2)

    return 1 ;

   s= print\_first\_n\_fibinocci\_terms(n-1) + print\_first\_n\_fibinocci\_terms(n-2) ;

    return s;

  }

Output

Enter number of terms :10

1 1 2 3 5 8 13 21 34 55

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

Code

#include<stdio.h>

int count\_digit\_in\_num(int);

int main()

{     int n;

     printf("Enter a number :");

     scanf(" %d",&n);

     printf("number of digit : %d",count\_digit\_in\_num(n));

    return 0;

}

int count\_digit\_in\_num(int n)

  {

     int s;

    if(n<10)

    return 1;

   s= count\_digit\_in\_num(n/10) + 1 ;

    //printf("%d",s);

    return s;

  }

Output

Enter a number :345

number of digit : 3

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

Code

#include<stdio.h>

int calculate\_power\_in\_num(int , int);

int main()

{     int p,b;

     printf("Enter a base :");

     scanf(" %d",&b);

     printf("Enter a power :");

     scanf(" %d",&p);

     printf("The value is : %d",calculate\_power\_in\_num(b,p));

    return 0;

}

int calculate\_power\_in\_num(int b , int p)

  {

     int s;

    if(p==1)

    return b;

   s= b\*calculate\_power\_in\_num(b,p-1) ;

    //printf("%d",s);

    return s;

  }

Output

Enter a base :3

Enter a power :3

The value is : 27