1. **Program to check whether entered number is palindrome or not using recursion.**

#include <stdio.h>

#include <math.h>

int return\_reverse(int number, int count){

if(number == 0){

return 0;

}

else{

return number%10 \* pow(10,count-1) + return\_reverse(number/10,count-1);

}

}

int main(){

int number,number\_copy;

int count = 0,reversed;

printf("Enter any number: ");

scanf("%d",&number);

number\_copy = number;

while(number > 0){

number /= 10;

count ++;

}

reversed = return\_reverse(number\_copy,count);

if(number\_copy == reversed){

printf("Palindrome !!");

}

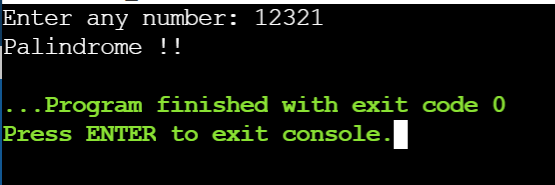
else{

printf("Not Palindrome !!");

}

return 0;

}

**OUTPUT:**

1. **Program to calculate power (ab) using recursion**

#include <stdio.h>

int pow(int a, int b){

if(b == 0){

return 1;

}

else{

return a \* pow(a, b - 1);

}

}

int main(){

int a,b,power;

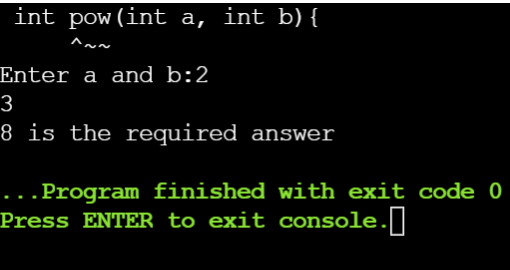
printf("Enter a and b:");

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

power = pow(a,b);

printf("%d is the required answer", power);

return 0;

**Output:**

1. **sum of digits of a number using recursion**

#include <stdio.h>

int sum(int number){

if(number == 0){

return 0;

}

else{

return number % 10 + sum(number / 10);

}

}

int main(){

int number, ans;

printf("Enter any number: ");

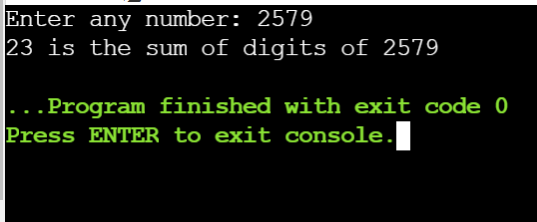
scanf("%d",&number);

ans = sum(number);

printf("%d is the sum of digits of %d",ans,number);

return 0;

}

**Output:**

1. //calculation of the factorial of an integer number using recursive function.

#include<stdio.h>

#include<conio.h>

void main(){

int n;

int sum\_natural(int);

printf("Enter the value of n:");

scanf("%d",&n);

printf("sum of %d natural numbers : %d",n,sum\_natural(n));

getch();

}

int sum\_natural(int n){

if(n==1)

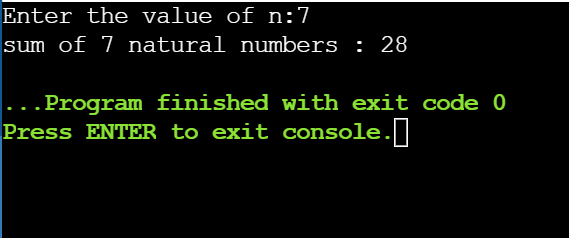
return 1;

else

return n + sum\_natural(n-1);

}

**Output:**



1. **Tower of Hanoi**

#include <stdio.h>

#include <string.h>

void honai(int n,char source, char auxillary, char destination){

if(n > 0){

honai(n-1,source, destination, auxillary);

printf("Move Disk %d from %c to %c\n",n,source,auxillary);

honai(n-1,destination,auxillary,source);

}

}

int main(){

int n = 3;

char source,auxillary,destination;

source = 'A';

auxillary = 'B';

destination = 'C';

honai(n,source, auxillary,destination);

return 0;

}

**Output:**

