**Program using Recursion**

1)Write a program in C to print the first 50 natural numbers using recursion.

void printNumbers(int n) {

if (n <= 50) {

printf("%d ", n);

printNumbers(n + 1);

}

}

#include<stdio.h>

int main()

{

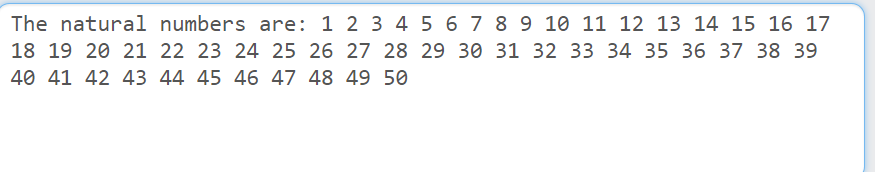
printf("The natural numbers are: ");

printNumbers(1);

printf("\n");

return 0;

}



**2.** Write a program in C to calculate the sum of numbers from 1 to n using recursion.

int calculateSum(int n)

{

if (n == 0)

return 0;

else

return n + calculateSum(n - 1);

}

#include <stdio.h>

int main()

{

int n, sum;

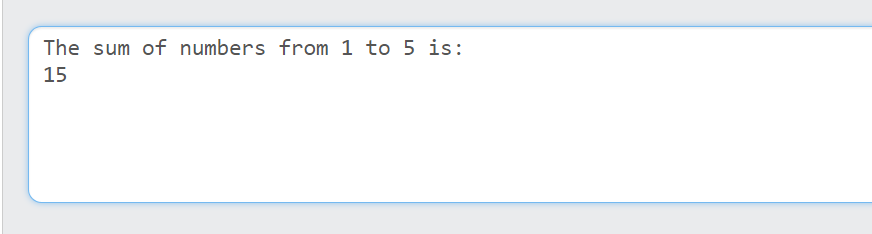
scanf("%d", &n);

sum = calculateSum(n);

printf("The sum of numbers from 1 to %d is: \n%d", n, sum);

return 0;

}



**3.** Write a program in C to print the Fibonacci Series using recursion.

#include<stdio.h>

int main()

{

int n,i,f;

scanf("%d",&n);

for(i=0;i<n;i++)

{

f=fibbanoci(i);

printf("%d ",f);

}

return 0;

}

int fibbanoci(int n)

{

int fib;

if(n==0)

fib=0;

else if(n==1)

fib=1;

else

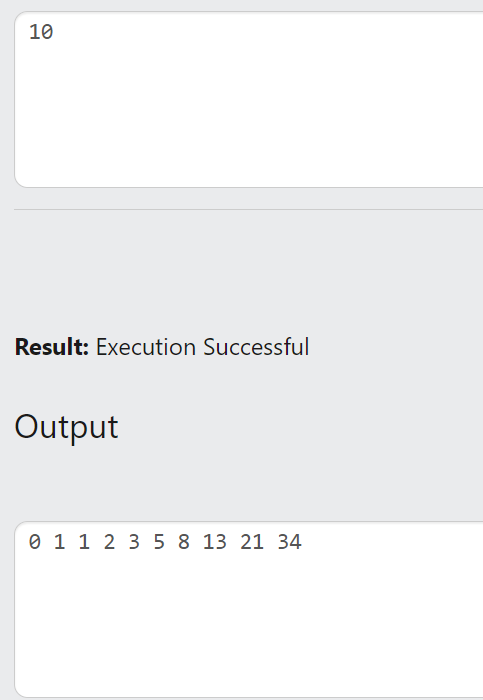
{

fib=fibbanoci(n-1)+fibbanoci(n-2);

}

return fib;

}



**4.** Write a program in C to print the array elements using recursion.

#include<stdio.h>

int main()

{

int n,i;

scanf("%d",&n);

int a[n];

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

print(a,n);

return 0;

}

void print(int a[],int n)

{

if(n<=0)

return;

else

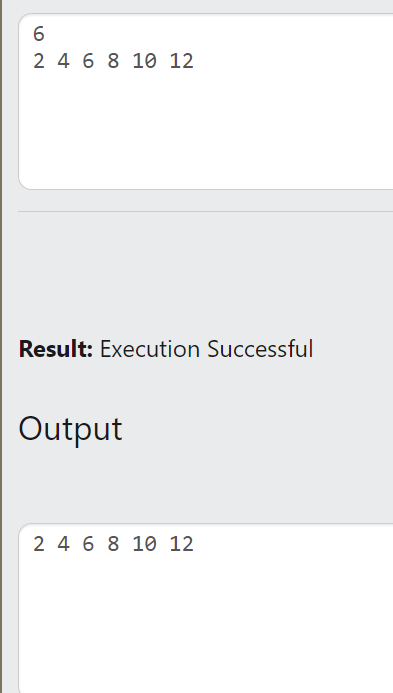
{

print(a,n-1);

printf("%d ",a[n-1]);

}

}



**5.** Write a program in C to print the array elements in reverse order using recursion.

void printArrayReverse(int arr[], int n,int index)

{

if (index < 0)

{

return;

}

printf("%d ", arr[index]);

printArrayReverse(arr,n, index - 1);

}

#include <stdio.h>

int main()

{

int n;

scanf("%d", &n);

int arr[n];

for (int i = 0; i < n; i++)

{

scanf("%d", &arr[i]);

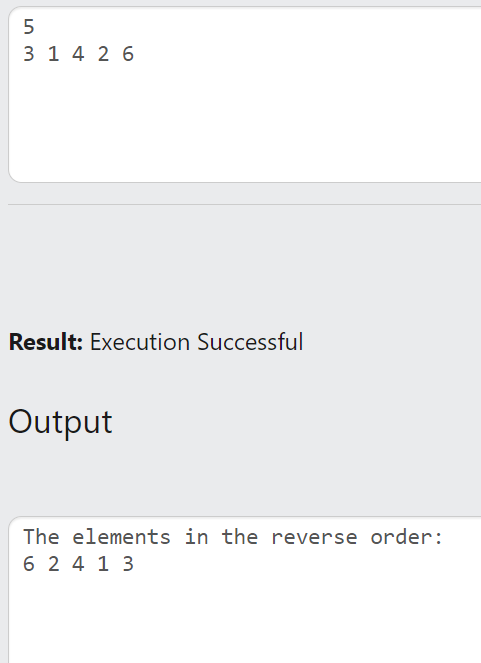
}

printf("The elements in the reverse order:\n");

printArrayReverse(arr,n, n- 1);

return 0;

}



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

int countDigits(int num)

{

if (num == 0)

{

return 0;

}

return 1 + countDigits(num / 10);

}

#include <stdio.h>

int main()

{

int number;

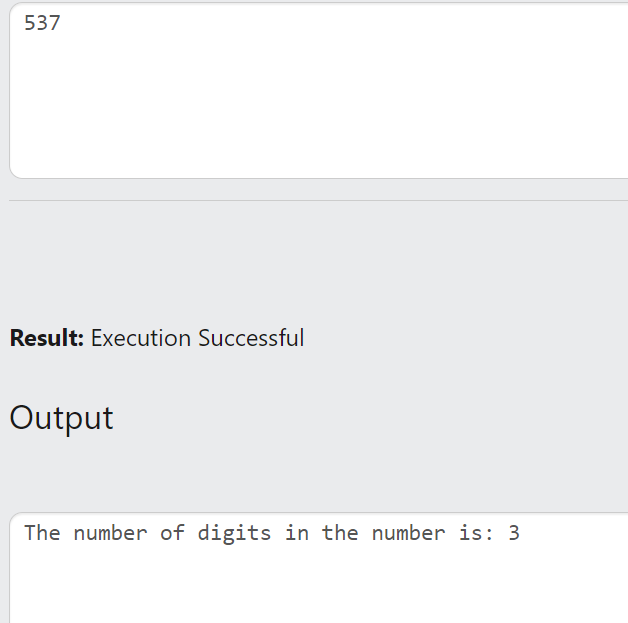
scanf("%d", &number);

int digitCount = countDigits(number);

printf("The number of digits in the number is: %d\n", digitCount);

return 0;

}



**7** Write a program in C to find the sum of digits of a number using recursion.   
  
#include <stdio.h>

int sumOfDigits(int num);

int main()

{

int num;

scanf("%d", &num);

int sum = sumOfDigits(num);

printf("The sum of digits of %d = %d\n", num, sum);

return 0;

}

int sumOfDigits(int num)

{

if (num == 0)

{

return 0;

}

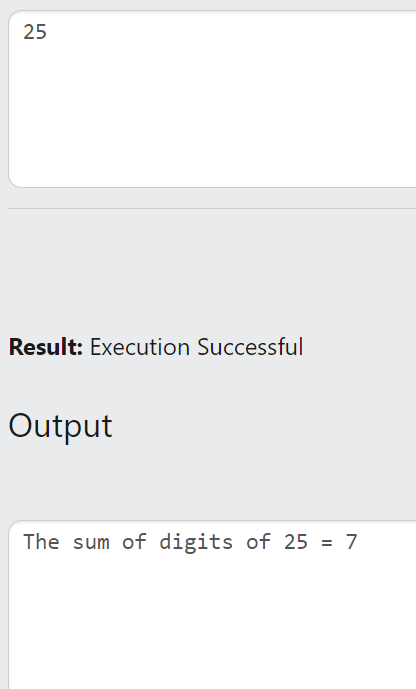
else

{

return (num % 10) + sumOfDigits(num / 10);

}

}



**8.** Write a program in C to get the largest element of an array using recursion.   
#include <stdio.h>

int findLargest(int arr[], int n);

int main()

{

int n;

scanf("%d", &n);

int arr[n];

for (int i = 0; i < n; i++)

{

scanf("%d", &arr[i]);

}

int largest = findLargest(arr, n);

printf("Largest element of the array is: %d\n", largest);

return 0;

}

int findLargest(int arr[], int n)

{

if (n == 1)

{

return arr[0];

}

else

{

int max = findLargest(arr, n - 1);

if (arr[n-1] > max)

{

return arr[n - 1];

}

else

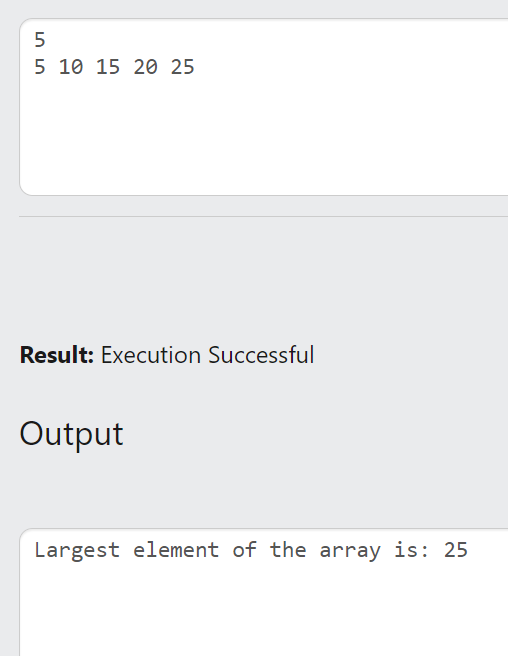
{

return max;

}

}

}



**9.** Write a program in C to reverse a string using recursion.

#include<stdio.h>

#include<string.h>

char s1[50];

int i=0;

void reverse(char s[],int len)

{

if(len==0)

puts(s1);

else

{

s1[i]=s[len-1];

i++;

reverse(s,len-1);

}

}

int main()

{

int len;

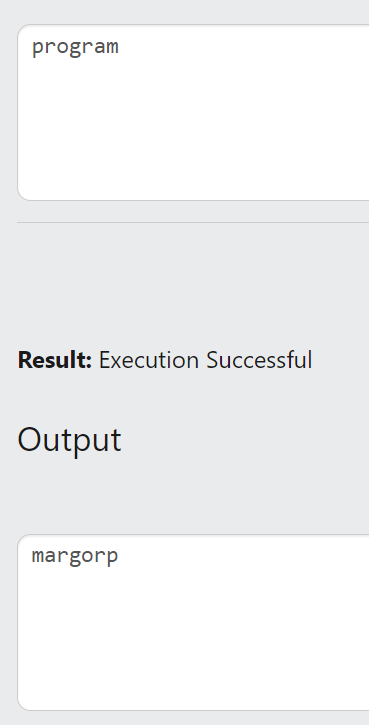
char s[50];

gets(s);

len=strlen(s);

reverse(s,len);

}



**10.** Write a program in C to find the Factorial of a number using recursion.   
  
#include <stdio.h>

int factorial(int n);

int main()

{

int num;

scanf("%d", &num);

int fact = factorial(num);

printf("The factorial of %d is: %d \n", num, fact);

return 0;

}

int factorial(int n)

{

if (n == 0)

{

return 1;

}

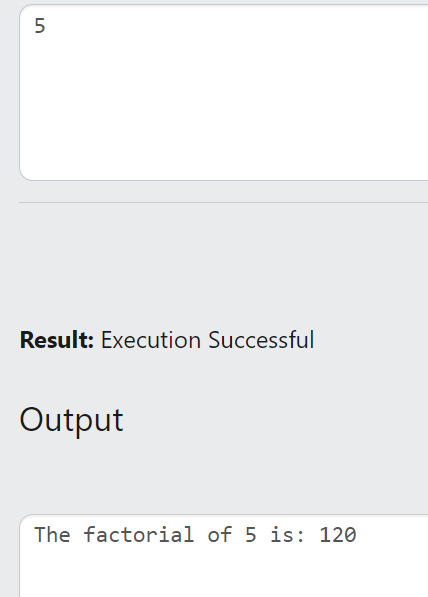
else

{

return n \* factorial(n - 1);

}

}



**11.** Write a program in C to convert a decimal number to binary using recursion.

#include<stdio.h>

void binary(n);

int main()

{

int n;

scanf("%d",&n);

binary(n);

}

void binary(n)

{

if (n==0)

return;

else

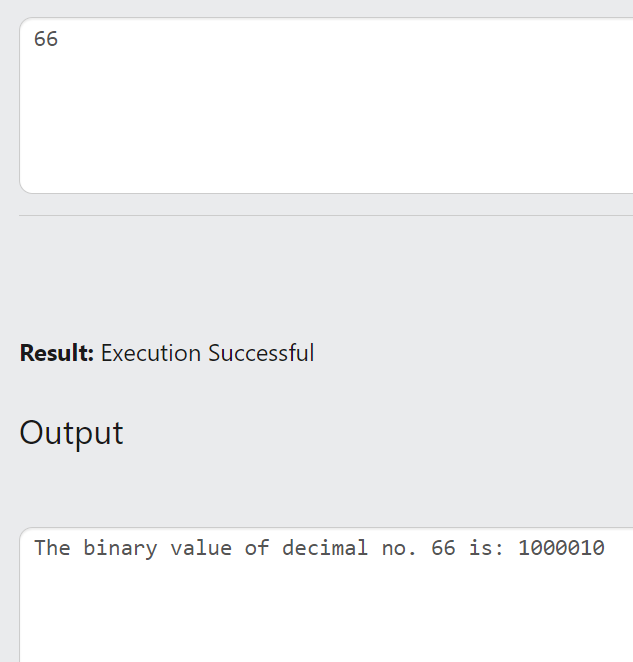
{

binary(n/2);

printf("%d",n%2);

}

}



**12.** Write a program in C to print even or odd numbers in a given range using recursion.

#include <stdio.h>

void printEven(int start, int end);

void printOdd(int start, int end);

int main()

{

int start, end;

scanf("%d", &end);

start = 1;

printf("All even numbers from %d to %d are: ", start, end);

printEven(start, end);

printf("\n");

printf("All odd numbers from %d to %d are: ", start, end);

printOdd(start, end);

printf("\n");

return 0;

}

void printEven(int start, int end)

{

if (start <= end)

{

if (start % 2 == 0)

{

printf("%d ", start);

}

printEven(start + 1, end);

}

}

void printOdd(int start, int end)

{

if (start <= end)

{

if (start % 2 != 0)

{

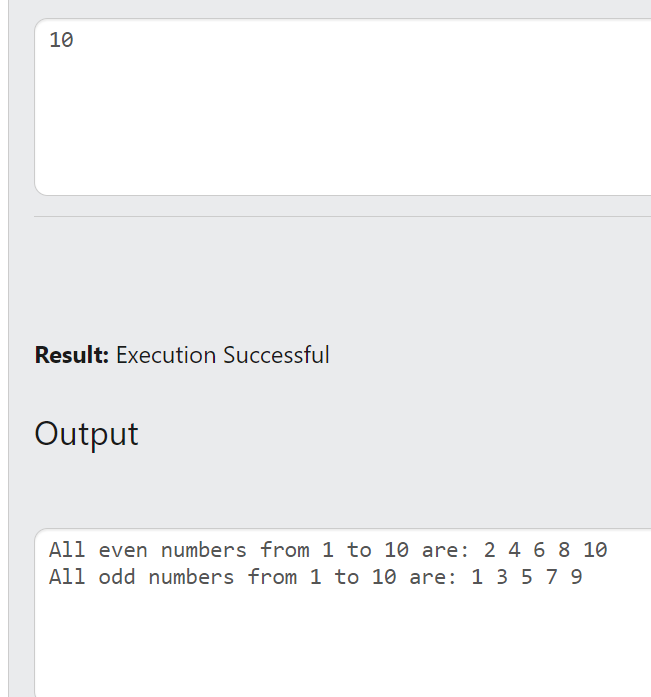
printf("%d ", start);

}

printOdd(start + 1, end);

}

}



**13.** Write a C program to check whether a given string is a palindrome or not using recursion.

#include<stdio.h>

#include<string.h>

char s1[50];

int i=0;

void reverse(char s[],int len)

{

if (len==0)

{

//puts(s1);

return;

}

else

{

s1[i]=s[len-1];

i++;

reverse(s,len-1);

}

}

int main()

{

int len;

char s[50];

gets(s);

len=strlen(s);

reverse(s,len);

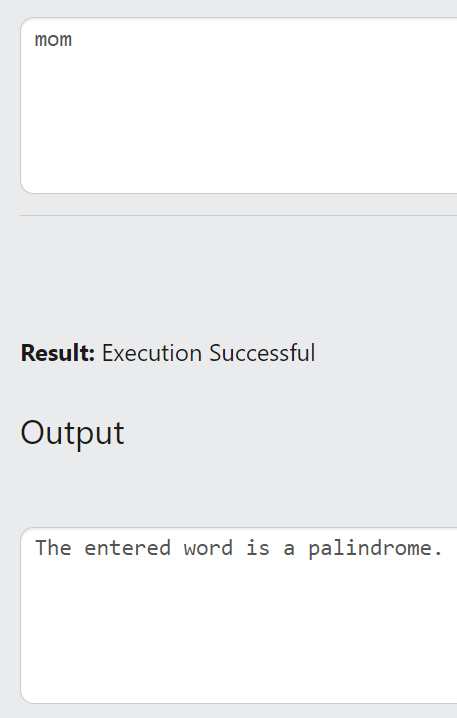
if (strcmp(s1,s)==0)

printf("The entered word is a palindrome. ");

else

printf("The entered word is not a palindrome. ");

}



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

#include<stdio.h>

int power(int a,int b)

{

if (b==0)

return 1;

else

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

}

int main()

{

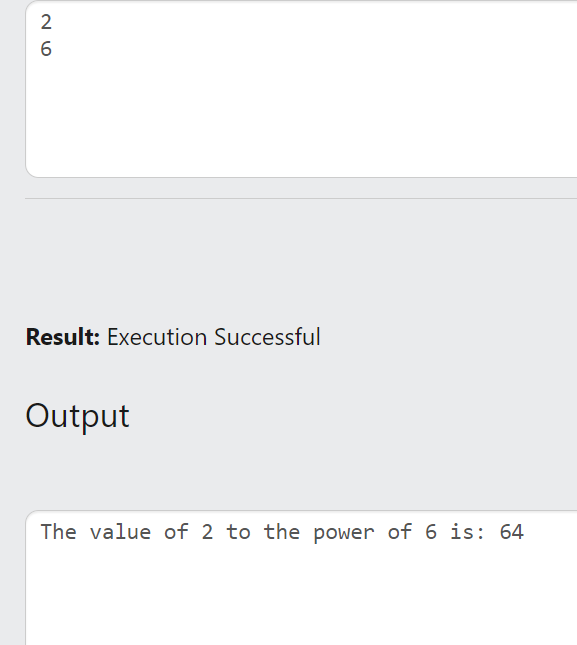
int a,b,v;

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

v=power(a,b);

printf("The value of %d to the power of %d is: %d",a,b,v);

}



**15.** Write a program in C to copy one string to another using recursion.

#include<stdio.h>

#include<string.h>

char s2[50];

void copy(char s1[],int n);

int main()

{

int len;

char s1[50];

gets(s1);

len=strlen(s1);

copy(s1,len);

printf("The first string is : %s\n",s1);

printf("The copied string is : %s",s2);

}

void copy (char s1[], int n)

{

if (n==0)

{

//puts(s2);

return;

}

else

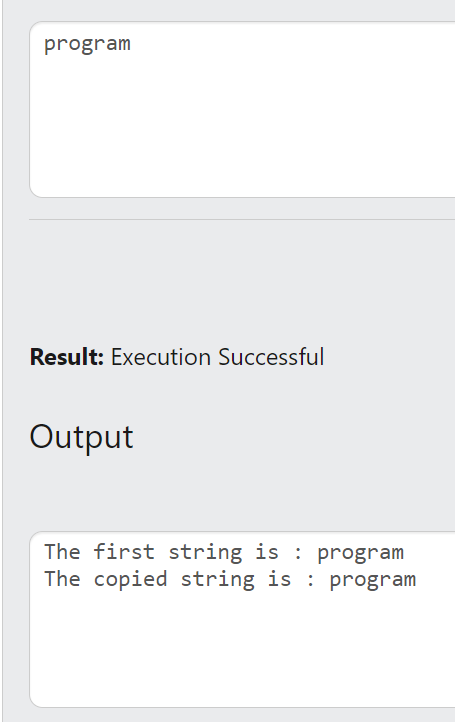
{

s2[n-1]=s1[n-1];

copy(s1,n-1);

}

}



**16)** C Program to Search an Element in an Array using Recursion. (Linear search)

#include<stdio.h>

int pos=-1;

int linearsearch(int a[],int n,int e)

{

if (n<0)

return pos;

else if(a[n-1]==e)

{

pos=n-1;

return pos;

}

else

{

linearsearch(a,n-1,e);

}

}

int main()

{

int n,i,e,s;

scanf("%d",&n);

int a[n];

for (i=0;i<n;i++)

scanf("%d",&a[i]);

scanf("%d",&e);

s=linearsearch(a,n,e);

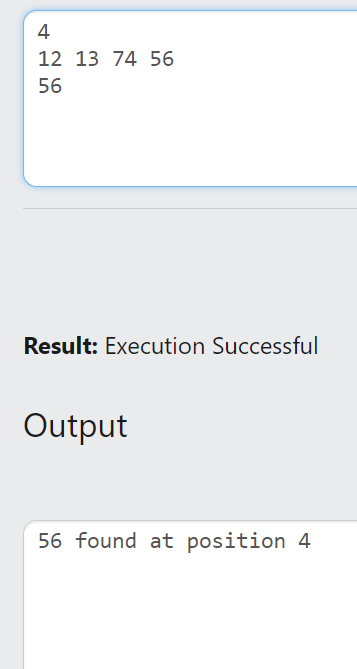
if (s==-1)

printf("Element Not Found");

else

printf("%d found at position %d",e,s+1);

}



**17)** C Program to Search an Element in an Array using Recursion. (Binary search)

#include<stdio.h>

int binarysearch(int a[],int f,int l,int e)

{

int mid;

mid=(f+l)/2;

if (f>l)

return -1;

else

{

if(a[mid]==e)

return mid;

else if(e<a[mid])

return binarysearch(a,f,mid-1,e);

else

return binarysearch(a,mid+1,l,e);

}

}

int main()

{

int n,i,e,s;

scanf("%d",&n);

int a[n];

for (i=0;i<n;i++)

scanf("%d",&a[i]);

scanf("%d",&e);

s=binarysearch(a,0,n,e);

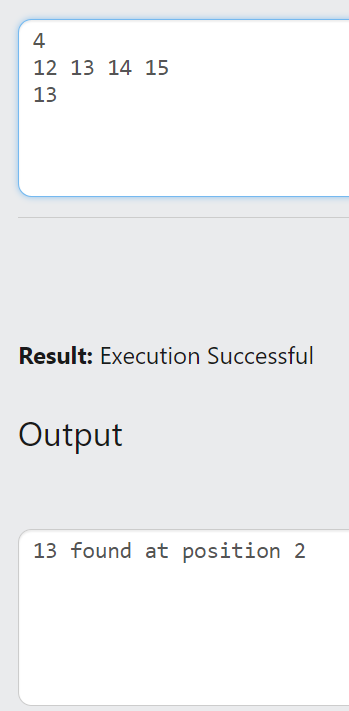
if (s==-1)

printf("Element Not Found");

else

printf("%d found at position %d",e,s+1);

}



Count of even numbers in a digit

#include<stdio.h>

int countofeven(int num)

{

if(num==0)

return 0;

int lastdigit=num%10;

int count=countofeven(num/10);

if(lastdigit%2==0)

{

count++;

}

return count;

}

int main()

{

int num;

scanf("%d",&num);

int count=countofeven(num);

printf("%d",count);

}

Display reverse of numbers within a range

void reverse(int n)

{

if(n<=0)

{

return;

}

else

{

printf("%d",n);

reverse(n-1);

}

}

#include<stdio.h>

int main()

{

int n;

scanf("%d",&n);

reverse(n);

return 0;

}

Display a digit in reverse order

void display(int n)

{

if(n==0)

{

return;

}

int num=n%10;

printf("%d",num);

display(n/10);

}

#include<stdio.h>

int main()

{

int n;

scanf("%d",&n);

display(n);

}