1. Write a C function to calculate NCR value

Code :

#include <stdio.h>

int nCr(int n, int r) {

if (r > n)

return 0;

if (r > n - r)

r = n - r;

int result = 1;

for (int i = 1; i <= r; i++) {

result = result \* (n - r + i) / i;

}

return result;

}

int main() {

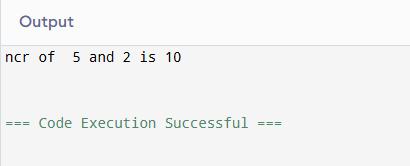
int n = 5, r = 2;

printf("ncr of %d and %d is %d\n", n, r, nCr(n, r));

return 0;

}

Output :



1. Write a C function to transpose of a matrix

Code :

#include <stdio.h>

int main() {

int rows, cols;

printf("Enter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

int original[rows][cols];

int transposed[cols][rows];

printf("Enter elements of the matrix:\n");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

printf("Element [%d][%d]: ", i, j);

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

}

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

transposed[j][i] = original[i][j];

}

}

printf("\nOriginal matrix:\n");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

printf("%d ", original[i][j]);

}

printf("\n");

}

printf("\nTransposed matrix:\n");

for (int i = 0; i < cols; i++) {

for (int j = 0; j < rows; j++) {

printf("%d ", transposed[i][j]);

}

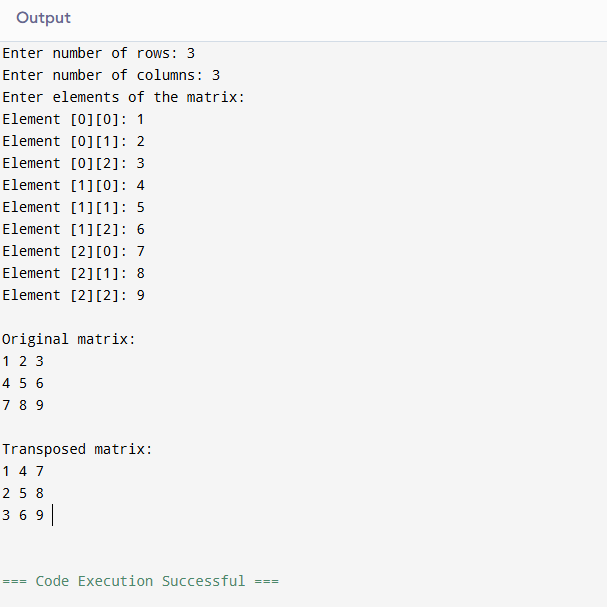
printf("\n");

}

return 0;

}

Output :



1. Write a recursive function to generate Fibonacci series

Code :

#include <stdio.h>

int fibonacci(int n) {

if (n == 0)

return 0;

else if (n == 1)

return 1;

else

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

}

int main() {

int n;

printf("Enter a number : ");

scanf("%d", &n);

printf("Fibonacci series: ");

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

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

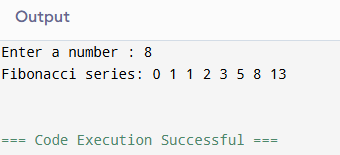
}

printf("\n");

return 0;

}

Output:



1. Write a recursive function to find the LCM of two numbers

Code :

#include <stdio.h>

int lcm(int a, int b, int multiple) {

if (multiple % a == 0 && multiple % b == 0)

return multiple;

return lcm(a, b, multiple + 1);

}

int main() {

int num1, num2;

printf("Enter two numbers: ");

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

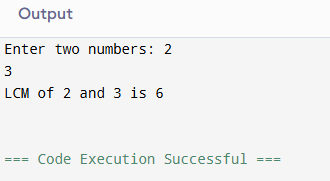
int start = (num1 > num2) ? num1 : num2;

printf("LCM of %d and %d is %d\n", num1, num2, lcm(num1, num2, start))

return 0;

}

Output :



1. Write a recursive function to find the GCD of two numbers

Code :

#include <stdio.h>

int gcd(int a, int b) {

if (b == 0)

return a;

return gcd(b, a % b);

}

int main() {

int num1, num2;

printf("Enter two numbers: ");

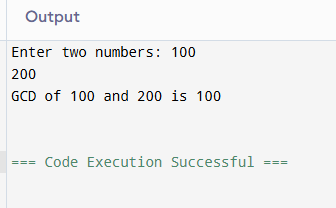
scanf("%d %d", &num1, &num2);

printf("GCD of %d and %d is %d\n", num1, num2, gcd(num1, num2));

return 0;

}

Output :



1. Write a recursive function to check whether the given number is palindrome or not

Code :

#include <stdio.h>

int reverseNumber(int num, int rev) {

if (num == 0)

return rev;

return reverseNumber(num / 10, rev \* 10 + num % 10);

}

int isPalindrome(int num) {

int reversed = reverseNumber(num, 0);

return (num == reversed);

}

int main() {

int number;

printf("Enter a number: ");

scanf("%d", &number);

if (isPalindrome(number))

printf("%d is a palindrome.\n", number);

else

printf("%d is NOT a palindrome.\n", number);

return 0;

}

Output :

