1. Write a program to Print Fibonacci Series using recursion

CODING:

#include <stdio.h>

int fibonacci(int n) {

if (n <= 1)

return n;

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

}

int main() {

int n, i;

printf("Enter the number of terms: ");

scanf("%d", &n);

printf("Fibonacci Series: ");

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

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

}

return 0;

}

1. Write a program to check the given no is Armstrong or not using recursive function

CODING:

#include <stdio.h>

int power(int base, int exponent);

int isArmstrong(int num);

int main() {

int number;

printf("Enter a number: ");

scanf("%d", &number);

if (isArmstrong(number))

printf("%d is an Armstrong number.\n");

else

printf("%d is not an Armstrong number.\n");

return 0;

}

int power(int base, int exponent) {

if (exponent == 0)

return 1;

return base \* power(base, exponent - 1);

}

int isArmstrong(int num) {

int originalNum, remainder, result = 0, n = 0;

originalNum = num;

// Count number of digits

while (originalNum != 0) {

originalNum /= 10;

++n;

}

originalNum = num;

// Check if the number is Armstrong

while (originalNum != 0) {

remainder = originalNum % 10;

result += power(remainder, n);

originalNum /= 10;

}

if (result == num)

return 1;

else

return 0;

}

1. Write a program to find the GCD of two numbers using recursive factorization

CODING:

#include <stdio.h>

int gcd(int a, int b) {

if (b == 0) {

return a;

} else {

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", num1, num2, gcd(num1, num2));

return 0;

}

1. Write a program to get the largest element of an array.

CODING:

#include <stdio.h>

int main() {

int arr[] = {10, 20, 4, 45, 99};

int n = sizeof(arr) / sizeof(arr[0]);

int max = arr[0];

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

if (arr[i] > max) {

max = arr[i];

}

}

printf("Largest element of the array is: %d", max);

return 0;

}

1. Write a program to find the Factorial of a number using recursion.

CODING:

#include <stdio.h>

unsigned long long factorial(unsigned int n);

int main() {

unsigned int number;

printf("Enter a non-negative integer: ");

scanf("%u", &number);

printf("Factorial of %u = %llu\n", number, factorial(number));

return 0;

}

unsigned long long factorial(unsigned int n) {

if (n == 0) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

1. Write a program for to copy one string to another using recursion

CODING:

#include <stdio.h>

void copyString(char \*source, char \*destination) {

if (\*source == '\0') {

\*destination = '\0';

return;

}

\*destination = \*source;

copyString(source + 1, destination + 1);

}

int main() {

char source[] = "Hello, World!";

char destination[50];

copyString(source, destination);

printf("Source String: %s\n", source);

printf("Copied String: %s\n", destination);

return 0;

}

7. Write a program to print the reverse of a string using recursion

CODING:

#include <stdio.h>

void reverseString(char str[]) {

if (str[0] == '\0') {

return;

}

reverseString(&str[1]);

printf("%c", str[0]);

}

int main() {

char str[] = "Hello, World!";

reverseString(str);

return 0;

}

8. Write a program to generate all the prime numbers using recursion

CODING:

#include <stdio.h>

int isPrime(int num, int i) {

if (i == 1) {

return 1;

} else {

if (num % i == 0) {

return 0;

} else {

return isPrime(num, i - 1);

}

}

}

void generatePrimes(int n) {

if (n > 1) {

if (isPrime(n, n / 2) == 1) {

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

}

generatePrimes(n - 1);

}

}

int main() {

int limit;

printf("Enter the limit to generate prime numbers: ");

scanf("%d", &limit);

printf("Prime numbers up to %d are:\n", limit);

generatePrimes(limit);

return 0;

}

9. Write a program to check a number is a prime number or not using recursion

CODING:

#include <stdio.h>

int isPrime(int num, int i) {

if (i == 1) {

return 1;

} else {

if (num % i == 0) {

return 0;

} else {

return isPrime(num, i - 1);

}

}

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num < 2) {

printf("%d is not a prime number.", num);

} else {

if (isPrime(num, num / 2) == 1) {

printf("%d is a prime number.", num);

} else {

printf("%d is not a prime number.", num);

}

}

return 0;

}

10. Write a program for to check whether a given String is Palindrome or not using recursion

CODING:

#include <stdio.h>

#include <string.h>

#include <stdbool.h>

bool isPalindrome(char str[], int start, int end) {

if (start >= end) {

return true;

}

if (str[start] != str[end]) {

return false;

}

return isPalindrome(str, start + 1, end - 1);

}

int main() {

char str[100];

printf("Enter a string: ");

scanf("%s", str);

int len = strlen(str);

if (isPalindrome(str, 0, len - 1)) {

printf("%s is a palindrome.\n", str);

} else {

printf("%s is not a palindrome.\n", str);

}

return 0;

}