1- Write a program to further calculate and display the product of digits if the entered positive integer 'num' is odd. If 'num' is even, the program should continue to display the message "No result."

اكتب برنامجًا لإجراء المزيد من الحساب وعرض حاصل ضرب الأرقام إذا كان العدد الصحيح الموجب "num" الذي تم إدخاله فرديًا. إذا كان الرقم 'num' زوجيًا، فيجب أن يستمر البرنامج في عرض الرسالة "No result.".

Input

```
Enter a positive integer: 1573
```

Output

```
Number of digits: 4
Sum of digits: 16
Product of digits: 105
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to count the number of digits in a positive integer recursively
int countDigits(int n) {
    if (n == 0)
        return 0;
    else
        return 1 + countDigits(n / 10);
}

// Function to calculate the sum of digits in a positive integer recursively
int sumDigits(int n) {
    if (n == 0)
        return 0;
    else
        return n % 10 + sumDigits(n / 10);
}

// Function to calculate the product of digits in a positive integer recursively
int productDigits(int n) {
    if (n == 0)
        return 1;
    else
        return n % 10 * productDigits(n / 10);
}

int main() {
    int num;

    // Input: Get a positive integer from the user
    cout << "Enter a positive integer: ";
    cin >> num;

    // Check if the number is odd
    if (num % 2) {
        // Output: Display the number of digits
        cout << "Number of digits: " << countDigits(num) << endl;

        // Output: Display the sum of digits
        cout << "Sum of digits: " << sumDigits(num) << endl;

        // Output: Display the product of digits
        cout << "Product of digits: " << productDigits(num) << endl;
    } else {
        cout << "No result." << endl;
    }

    return 0;
}
```

2- Extend the previous program to find and display the reverse of the entered positive integer 'num' if it is odd. If 'num' is even, the program should continue to display the message "No result." using recursion

قم بتوسيع البرنامج السابق للعثور على عكس العدد الصحيح الموجب "num" الذي تم إدخاله وعرضه إذا كان فرديًا. إذا كان الرقم 'num' زوجيًا، فيجب أن يستمر البرنامج في عرض الرسالة "لا توجد نتيجة". using recursion

Input

```
Enter a positive integer: 12345
```

Output

```
Number of digits: 5
Sum of digits: 15
Product of digits: 120
Reverse of digits: 54321
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to count the number of digits in a positive integer recursively
int countDigits(int n) {
    if (n == 0)
        return 0;
    else
        return 1 + countDigits(n / 10);
}

// Function to calculate the sum of digits in a positive integer recursively
int sumDigits(int n) {
    if (n == 0)
        return 0;
    else
        return n % 10 + sumDigits(n / 10);
}

// Function to calculate the product of digits in a positive integer recursively
int productDigits(int n) {
    if (n == 0)
        return 1;
    else
        return n % 10 * productDigits(n / 10);
}

// Function to reverse the digits of a positive integer recursively
int reverseDigits(int n, int reversedNum) {
    if (n == 0)
        return reversedNum;
    else
        return reverseDigits(n / 10, reversedNum * 10 + n % 10);
}

int main() {
    int num;

    // Input: Get a positive integer from the user
    cout << "Enter a positive integer: ";
    cin >> num;

    // Check if the number is odd
    if (num % 2) {
        // Output: Display the number of digits
        cout << "Number of digits: " << countDigits(num) << endl;

        // Output: Display the sum of digits
        cout << "Sum of digits: " << sumDigits(num) << endl;

        // Output: Display the product of digits
        cout << "Product of digits: " << productDigits(num) << endl;

        // Output: Display the reverse of digits
        cout << "Reverse of digits: " << reverseDigits(num, 0) << endl;
    } else {
        cout << "No result." << endl;
    }

    return 0;
}
```

3- Write a program that takes a positive integer 'num' as input from the user. The program should determine and display whether 'num' is a prime number or not using recursion. .

اكتب برنامجًا يأخذ عددًا صحيحًا موجبًا "num" كمدخل من المستخدم. يجب أن يحدد البرنامج ويعرض ما إذا كان "num" رقمًا أوليًا أم لا يستخدم التكرار. .

Input

```
Enter a positive integer: 5
```

Output

```
5 is a prime number.
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
#include <cmath>
using namespace std;

// Function to check if a number is prime recursively
bool isPrime(int n, int i = 2) {
    // Base cases
    if (n <= 1)
        return false;
    if (i > sqrt(n))
        return true;

    // Check for divisibility
    if (n % i == 0)
        return false;

    // Recursive call with incremented divisor
    return isPrime(n, i + 1);
}

int main() {
    int num;

    // Input: Get a positive integer from the user
    cout << "Enter a positive integer: ";
    cin >> num;

    // Output: Display whether the number is prime or not
    if (isPrime(num))
        cout << num << " is a prime number." << endl;
    else
        cout << num << " is not a prime number." << endl;

    return 0;
}
```

4- Write a program to find the sum of natural numbers up to a given number using recursion.

اكتب برنامجًا لإيجاد مجموع الأعداد الطبيعية حتى رقم معين using recursion.

Input

```
Enter a number: 5
```

Output

```
Sum of natural numbers up to 5 is: 15
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int sumOfNaturalNumbers(int n) {
    if (n == 0)
        return 0;
    else
        return n + sumOfNaturalNumbers(n - 1);
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    cout << "Sum of natural numbers up to " << num << " is: " << sumOfNaturalNumbers(num) << endl;
    return 0;
}
```

5- Write a program to calculate the power of a number using recursion.

اكتب برنامجًا لحساب قوة الرقم using recursion.

Input

```
Enter the base: 2
Enter the exponent: 5
```

Output

```
2 raised to the power 5 is: 32
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int power(int base, int exponent) {
    if (exponent == 0)
        return 1;
    else
        return base * power(base, exponent - 1);
}

int main() {
    int base, exponent;
    cout << "Enter the base: ";
    cin >> base;
    cout << "Enter the exponent: ";
    cin >> exponent;

    cout << base << " raised to the power " << exponent << " is: " << power(base, exponent) << endl;
    return 0;
}
```

6- Write a program to find the Greatest Common Divisor Greatest Common Divisorof two numbers using recursion.

اكتب برنامجًا لإيجاد القاسم المشترك الأكبر القاسم المشترك الأكبر لعددين using recursion.

Input

```
Enter two numbers: 24 36
```

Output

```
GCD of 24 and 36 is: 12
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int gcd(int a, int b) {
    if (b == 0)
        return a;
    else
        return gcd(b, a % b);
}

int main() {
    int num1, num2;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    cout << "GCD of " << num1 << " and " << num2 << " is: " << gcd(num1, num2) << endl;
    return 0;
}
```

7- Write a program to reverse a string using recursion.

اكتب برنامجًا لعكس string بـusing recursion.

Input

```
Enter a string: hello
```

Output

```
Reversed string: olleh
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
#include <cstring>
using namespace std;

void reverseString(char str[], int start, int end) {
    if (start < end) {
        swap(str[start], str[end]);
        reverseString(str, start + 1, end - 1);
    }
}

int main() {
    char str[100];
    cout << "Enter a string: ";
    cin.getline(str, 100);

    reverseString(str, 0, strlen(str) - 1);

    cout << "Reversed string: " << str << endl;
    return 0;
}
```

8- Write a program to check if a given string is a palindrome using recursion.

اكتب برنامجًا للتحقق مما إذا كانت string المعطاة متناظرة using recursion.

Input

```
Enter a string: lmml
```

Output

```
The string is a palindrome.
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
#include <cstring>
using namespace std;

bool isPalindrome(const 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];
    cout << "Enter a string: ";
    cin.getline(str, 100);

    int len = strlen(str);
    if (isPalindrome(str, 0, len - 1))
        cout << "The string is a palindrome." << endl;
    else
        cout << "The string is not a palindrome." << endl;

    return 0;
}
```

9- Write a program to generate the Fibonacci series up to a given number using recursion.

اكتب برنامجًا لتوليد Fibonacci series حتى رقم معين using recursion.

Input

```
Enter the number of terms: 8
```

Output

```
Fibonacci Series up to 8: 0 1 1 2 3 5 8 13
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void fibonacciSeries(int a, int b, int n) {
    if (n > 0) {
        cout << a << " ";
        fibonacciSeries(b, a + b, n - 1);
    }
}

int main() {
    int num;
    cout << "Enter the number of terms: ";
    cin >> num;

    cout << "Fibonacci Series up to " << num << ": ";
    fibonacciSeries(0, 1, num);
    cout << endl;

    return 0;
}
```

10- Write a program to find the sum of elements in an array using recursion.

اكتب برنامجًا لإيجاد مجموع العناصر في array ب using recursion.

Input

```
Enter the size of the array: 5
Enter the elements of the array: 1 2 3 4 5
```

Output

```
Sum of array elements: 15
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int sumOfArray(int arr[], int n) {
    if (n == 0)
        return 0;
    else
        return arr[n - 1] + sumOfArray(arr, n - 1);
}

int main() {
    int n;
    cout << "Enter the size of the array: ";
    cin >> n;

    int arr[n];
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    cout << "Sum of array elements: " << sumOfArray(arr, n) << endl;
    return 0;
}
```
