
1- Write a program to convert a decimal number to binary using recursion.

كتابة برنامج لتحويل الرقم العشري إلى ثنائي using recursion.

Input

```
Enter a decimal number: 10
```

Output

```
Binary representation: 1010
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void decimalToBinary(int n) {
    if (n > 0) {
        decimalToBinary(n / 2);
        cout << n % 2;
    }
}

int main() {
    int num;
    cout << "Enter a decimal number: ";
    cin >> num;

    cout << "Binary representation: ";
    decimalToBinary(num);
    cout << endl;

    return 0;
}
```

2- Write a program to check if a given positive integer is a palindrome using recursion.

اكتب برنامجًا للتحقق مما إذا كان هو palindrome ب using recursion

Input

```
Enter a positive integer: 1221
```

Output

```
1221 is a palindrome number.
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

bool isPalindromeNumber(int n, int &temp) {
    if (n == 0)
        return true;

    if (isPalindromeNumber(n / 10, temp) && (n % 10 == temp % 10)) {
        temp /= 10;
        return true;
    }

    return false;
}

int main() {
    int num;
    cout << "Enter a positive integer: ";
    cin >> num;

    int temp = num;
    if (isPalindromeNumber(num, temp))
        cout << num << " is a palindrome number." << endl;
    else
        cout << num << " is not a palindrome number." << endl;

    return 0;
}
```

3- Write a program to print the first N terms of a sequence using recursion.

اكتب برنامجًا لطباعة

N terms of a sequence using recursion.

Input

```
Enter the number of terms: 5
```

Output

```
First 5 terms of the sequence: 1 3 5 7 9
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void printSequence(int n, int term = 1) {
    if (n > 0) {
        cout << term << " ";
        printSequence(n - 1, term + 2);
    }
}

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

    cout << "First " << num << " terms of the sequence: ";
    printSequence(num);
    cout << endl;

    return 0;
}
```

4- Write a program to find the sum of even numbers from 2 to N using recursion.

اكتب برنامجًا لإيجاد مجموع الأعداد الزوجية من 2 إلى N.

Input

```
Enter a number: 8
```

Output

```
Sum of even numbers from 2 to 8: 20
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int sumOfEvenNumbers(int n) {
    if (n <= 0)
        return 0;
    else
        return (n % 2 == 0) ? n + sumOfEvenNumbers(n - 2) : sumOfEvenNumbers(n - 1);
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    cout << "Sum of even numbers from 2 to " << num << ": " << sumOfEvenNumbers(num) << endl;

    return 0;
}
```

5- Write a program to print each character of a string using recursion.

اكتب برنامجًا لطباعة كل حرف من string.

Input

```
Enter a string: Hello world
```

Output

```
Characters in the string: H e l l o   w o r l d
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void printCharacters(const char* str) {
    if (*str != '\0') {
        cout << *str << " ";
        printCharacters(str + 1);
    }
}

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

    cout << "Characters in the string: ";
    printCharacters(str);
    cout << endl;

    return 0;
}
```

6- Write a program to reverse the elements of an array using recursion.

اكتب برنامجًا لعكس عناصر array using recursion.

Input

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

Output

```
Reversed array: 5 4 3 2 1
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void reverseArray(int arr[], int start, int end) {
    if (start < end) {
        swap(arr[start], arr[end]);
        reverseArray(arr, start + 1, end - 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];

    reverseArray(arr, 0, n - 1);

    cout << "Reversed array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;

    return 0;
}
```

7- Write a program to check if a given number is a power of two using recursion.

اكتب برنامجًا للتحقق مما إذا كان رقم معين يمثل power of two .

Input

```
Enter a number: 15
```

Output

```
15 is not a power of two.
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

bool isPowerOfTwo(int n) {
    if (n == 0)
        return false;
    else if (n == 1)
        return true;
    else if (n % 2 != 0)
        return false;
    else
        return isPowerOfTwo(n / 2);
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    if (isPowerOfTwo(num))
        cout << num << " is a power of two." << endl;
    else
        cout << num << " is not a power of two." << endl;

    return 0;
}
```

8- Write a program that calculates the power of each element in an array using recursion. Given an array arr of size n, raise each element to the power of its index and store the results in a new array.

اكتب برنامجًا يحسب power of each element في array باستخدام recursion. نظرًا لarray بحجم n، ارفع كل عنصر إلى قوة فهرسه وقم بتخزين النتائج في array جديدة.

Input & Output

```
Enter the size of the array: 4
Enter the elements of the array: 1 2 3 4
Original Array: 1 2 3 4
Powers of each element: 1 2 9 64
```

Solution

```cpp
// www.gammal.tech

#include <iostream>

using namespace std;

void calculatePowers(int arr[], int powers[], int size, int index = 0) {
    if (index == size) {
        return;
    }

    int power = 1;
    for (int i = 0; i < index; ++i) {
        power *= arr[index];
    }
    powers[index] = power;

    calculatePowers(arr, powers, size, index + 1);
}

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

    int arr[size];
    int powers[size];

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

    calculatePowers(arr, powers, size);

    cout << "Original Array: ";
    for (int i = 0; i < size; ++i) {
        cout << arr[i] << " ";
    }
    cout << endl;

    cout << "Powers of each element: ";
    for (int i = 0; i < size; ++i) {
        cout << powers[i] << " ";
    }
    cout << endl;

    return 0;
}
```

9- Write a program that calculates the product of elements at even positions in an array using recursion. Given an array arr of size n, compute the product of elements at positions 0, 2, 4, etc., and store the result.

اكتب برنامجًا يحسب حاصل ضرب العناصر في مواضع زوجية في array باستخدام recursion. بالنظر إلى array بحجم n، قم بحساب ضرب العناصر في المواضع 0، 2، 4، وما إلى ذلك، وقم بتخزين النتيجة.

Input

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

Output

```
Product of elements at even positions: 15
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int calculateProduct(int arr[], int index, int size) {
    if (index >= size)
        return 1;

    return arr[index] * calculateProduct(arr, index + 2, size);
}

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];

    int product = calculateProduct(arr, 0, n);

    cout << "Product of elements at even positions: " << product << endl;

    return 0;
}
```

10- Write a program that counts the number of odd elements in an array using recursion. Given an array arr of size n, find and print the count of odd numbers.

اكتب برنامجًا يحسب عدد العناصر الفردية في array باستخدام recursion.
بالنظر إلى array بحجم n، ابحث عن عدد الأرقام الفردية واطبعه.

Input

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

Output

```
Number of odd elements in the array: 3
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

int countOddNumbers(int arr[], int index, int size) {
    if (index >= size)
        return 0;

    return (arr[index] % 2 != 0) + countOddNumbers(arr, index + 1, size);
}

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];

    int oddCount = countOddNumbers(arr, 0, n);

    cout << "Number of odd elements in the array: " << oddCount << endl;

    return 0;
}
```