

1- 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

```
// 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;
}
```

2- 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

```
// 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[i];
    }
    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;
}
```

3- 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

```
// 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;
}
```

4- 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

```
// 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;
}
```

5- 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

```

// 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;
}
```

---

6- 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

```

// 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;
}
```



7- 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

```
// 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;
}
```

8- 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

```
// 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;
}
```

9- Write a program that dynamically allocates memory for a float array of user-defined size, takes user input for each element, calculates the average, and displays it.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا ل float array ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويحسب المتوسط، ويعرضه.

## Input

```
Enter the size of the array: 3
Enter 3 float numbers:
1.2 0.5 3.6
```

## Output

```
Average of array elements: 1.77
```

## Solution

```
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>

int main() {
    float *arr;
    int size;

    // Get array size from user
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    arr = (float*)malloc(size * sizeof(float));

    // Take user input for array elements
    printf("Enter %d float numbers:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%f", &arr[i]);
    }

    // Calculate average
    float sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    float average = sum / size;

    // Display the average
    printf("Average of array elements: %.2f\n", average);

    // Free allocated memory
    free(arr);

    return 0;
}
```

---

10- Write a program that dynamically allocates memory for a character array of user-defined size, takes user input for the string, reverses it, and displays the reversed string.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ array أحرف ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لـ string ، ويعكسها، ويعرض string المعكوسة.

Input

```
Enter the size of the string: 6
Enter a string: gammal
```

Output

```
Reversed String: lammag
```

Solution

```
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    char *str;
    int size;

    // Get string size from user
    printf("Enter the size of the string: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    str = (char*)malloc((size + 1) * sizeof(char));

    // Take user input for string
    printf("Enter a string: ");
    scanf("%s", str);

    // Reverse the string
    int i, j;
    char temp;
    for (i = 0, j = size - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }

    // Display the reversed string
    printf("Reversed String: %s\n", str);

    return 0;
}
```