

1- Write a C++ program that demonstrates the use of dynamic memory allocation for a single floating-point variable. Allocate memory using new to store a floating-point value, assign a value to it, and then deallocate the memory using delete. Finally, display the value of the variable.

اكتب برنامج ++C يوضح استخدام تخصيص الذاكرة الديناميكية لمتغير floating-point واحد. تخصيص الذاكرة باستخدام جديد لتخزين قيمة floating-point ، وتعيين قيمة لها، ثم إلغاء تخصيص الذاكرة باستخدام الحذف. وأخيرا، عرض قيمة المتغير.

Output

Value: 42.5

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    // Allocate memory for a single floating-point variable
    float* numPtr = new float;

    // Assign a value to the variable
    *numPtr = 42.5;

    // Display the value of the variable
    cout << "Value: " << *numPtr << endl;

    // Deallocate the memory
    delete numPtr;

    return 0;
}
```

2- Write a C++ program that demonstrates the use of dynamic memory allocation for a single character variable. Allocate memory using new to store a character, assign a value to it, and then deallocate the memory using delete. Finally, display the value of the character.

اكتب برنامج ++C يوضح استخدام تخصيص الذاكرة الديناميكية لمتغير حرف واحد. تخصيص الذاكرة باستخدام جديد لتخزين حرف، وتعيين قيمة له، ثم إلغاء تخصيص الذاكرة باستخدام الحذف. وأخيرًا، قم بعرض قيمة الحرف.

Output

```
Value: A
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    // Allocate memory for a single character variable
    char* charPtr = new char;

    // Assign a value to the variable
    *charPtr = 'A';

    // Display the value of the character
    cout << "Value: " << *charPtr << endl;

    // Deallocate the memory
    delete charPtr;

    return 0;
}
```

3- Write a C++ program that demonstrates the use of dynamic memory allocation for an array of integers. Allocate memory using new to store integers, populate the array with user inputs, and then deallocate the memory using delete. Display the array elements before deallocating the memory.

اكتب برنامج ++C يوضح استخدام تخصيص الذاكرة الديناميكية لمجموعة من الأعداد الصحيحة. تخصيص الذاكرة باستخدام جديد لتخزين الأعداد الصحيحة، وملء array بإدخالات المستخدم، ثم إلغاء تخصيص الذاكرة باستخدام الحذف. عرض عناصر array قبل إلغاء تخصيص الذاكرة.

Input

```
Enter the size of the array: 4
Enter 4 integers:
1 2 3 4
```

Output

```
Array elements: 1 2 3 4
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

int main() {
    // Prompt the user for the size of the array
    int arraySize;
    cout << "Enter the size of the array: ";
    cin >> arraySize;

    // Allocate memory for an array of integers
    int* intArray = new int[arraySize];

    // Populate the array with user inputs
    cout << "Enter " << arraySize << " integers:" << endl;
    for (int i = 0; i < arraySize; ++i) {
        cin >> intArray[i];
    }

    // Display the array elements
    cout << "Array elements: ";
    for (int i = 0; i < arraySize; ++i) {
        cout << intArray[i] << " ";
    }

    // Deallocate the memory
    delete[] intArray;

    return 0;
}
```

4- Write a program to count the number of digits in a positive integer using recursion.

اكتب برنامج لحساب عدد الأرقام في عدد صحيح موجب using recursion.

Input

```
Enter a positive integer: 1234
```

Output

```
Number of digits: 4
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

int countDigits(int n) {
    if (n == 0)
        return 0;
    else
        return 1 + countDigits(n / 10);
}

int main() {
    int num;
    cout << "Enter a positive integer: ";
    cin >> num;
    cout << "Number of digits: " << countDigits(num) << endl;
    return 0;
}
```

5- Write a program that takes a positive integer input 'num' from the user and counts the number of digits in it. Implement this functionality using a recursive function named countDigits. Ensure that the program only performs the count if the entered number is odd; otherwise, it should display "no result."

اكتب برنامجًا يأخذ عددًا صحيحًا موجبًا "num" من المستخدم ويحسب عدد الأرقام فيه. Implement this functionality using a recursive function named countDigits. التأكيد من أن البرنامج لا يقوم بالعد إلا إذا كان الرقم المدخل فرديًا؛ وإلا، فيجب عرض "لا يوجد نتيجة".

Input

```
Enter a positive integer: 333
```

Output

```
Number of digits: 3
```

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);
}

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;
    } else {
        cout << "No result." << endl;
    }

    return 0;
}
```

6- Write a program to find the sum of digits of a positive integer using recursion.

اكتب برنامجًا لإيجاد مجموع أرقام عدد صحيح موجب using recursion.

Input

```
Enter a positive integer: 159
```

Output

```
Sum of digits is: 15
```

Solution

```

// www.gammal.tech

#include <iostream>
using namespace std;

int sumOfDigits(int n) {
    if (n == 0)
        return 0;
    else
        return n % 10 + sumOfDigits(n / 10);
}

int main() {
    int num;
    cout << "Enter a positive integer: ";
    cin >> num;
    cout << "Sum of digits is: " << sumOfDigits(num) << endl;
    return 0;
}
```

7- Write a program that prompts the user to enter a positive integer 'num'. The program should count and display the number of even digits in 'num'

اكتب برنامجًا يطلب من المستخدم إدخال عدد صحيح موجب "num". يجب أن يقوم البرنامج بحساب وعرض عدد الأرقام الزوجية في "num"

Input

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

Output

```
Number of even digits: 2
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to count the number of even digits in a positive integer recursively
int countEvenDigit(int n) {
    if (n == 0)
        return 0;
    else {
        int lastDigit = n % 10;
        // Check if the last digit is even
        if (lastDigit % 2 == 0)
            return 1 + countEvenDigit(n / 10);
        else
            return countEvenDigit(n / 10);
    }
}

int main() {
    int num;

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

    // Output: Display the number of even digits
    cout << "Number of even digits: " << countEvenDigit(num) << endl;

    return 0;
}
```

8- 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

```
// 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;
}
```

9- Write a program to find the Greatest Common Divisor
Greatest Common Divisor of two numbers using recursion.

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

Input

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

Output

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

Solution

```
// 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;
}
```

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

اكتب برنامجاً لعكس string بـ using recursion.

Input

```
Enter a string: hello
```

Output

```
Reversed string: olleh
```

Solution

```

● ● ●
// 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;
}
```
