**1. Hello World**

**Question:** Write a program to print "Hello World".

#include <iostream>

int main() {

std::cout << "Hello World" << std::endl;

return 0;

}

**2. Sum of Two Numbers**

**Question:** Write a program to add two numbers.

#include <iostream>

int main() {

int a = 5, b = 10;

int sum = a + b;

std::cout << "Sum: " << sum << std::endl;

return 0;

}

**3. Find the Largest Number**

**Question:** Write a program to find the largest of three numbers.

#include <iostream>

int main() {

int a = 10, b = 20, c = 30;

int largest = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

std::cout << "Largest: " << largest << std::endl;

return 0;

}

**4. Check Even or Odd**

**Question:** Write a program to check if a number is even or odd.

#include <iostream>

int main() {

int num = 5;

if (num % 2 == 0)

std::cout << "Even" << std::endl;

else

std::cout << "Odd" << std::endl;

return 0;

}

**5. Fibonacci Series**

**Question:** Write a program to generate Fibonacci series up to n terms.

#include <iostream>

int main() {

int n = 10, t1 = 0, t2 = 1, nextTerm;

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

std::cout << t1 << ", ";

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

}

return 0;

}

**6. Factorial of a Number**

**Question:** Write a program to find the factorial of a number.

#include <iostream>

int main() {

int n = 5;

long factorial = 1;

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

factorial \*= i;

}

std::cout << "Factorial: " << factorial << std::endl;

return 0;

}

**7. Prime Number Check**

**Question:** Write a program to check if a number is prime.

#include <iostream>

int main() {

int n = 29;

bool isPrime = true;

if (n <= 1)

isPrime = false;

for (int i = 2; i <= n / 2; ++i) {

if (n % i == 0) {

isPrime = false;

break;

}

}

if (isPrime)

std::cout << "Prime" << std::endl;

else

std::cout << "Not Prime" << std::endl;

return 0;

}

**8. Reverse a Number**

**Question:** Write a program to reverse a number.

#include <iostream>

int main() {

int n = 1234, reversed = 0;

while (n != 0) {

int digit = n % 10;

reversed = reversed \* 10 + digit;

n /= 10;

}

std::cout << "Reversed: " << reversed << std::endl;

return 0;

}

**9. Palindrome Check**

**Question:** Write a program to check if a number is a palindrome.

#include <iostream>

int main() {

int n = 121, original = n, reversed = 0;

while (n != 0) {

int digit = n % 10;

reversed = reversed \* 10 + digit;

n /= 10;

}

if (original == reversed)

std::cout << "Palindrome" << std::endl;

else

std::cout << "Not Palindrome" << std::endl;

return 0;

}

**10. Sum of Digits**

**Question:** Write a program to find the sum of digits of a number.

#include <iostream>

int main() {

int n = 1234, sum = 0;

while (n != 0) {

sum += n % 10;

n /= 10;

}

std::cout << "Sum of Digits: " << sum << std::endl;

return 0;

}

**11. Armstrong Number**

**Question:** Write a program to check if a number is an Armstrong number.

#include <iostream>

#include <cmath>

int main() {

int n = 153, original = n, sum = 0;

while (n != 0) {

int digit = n % 10;

sum += std::pow(digit, 3);

n /= 10;

}

if (original == sum)

std::cout << "Armstrong" << std::endl;

else

std::cout << "Not Armstrong" << std::endl;

return 0;

}

**12. Swap Two Numbers**

**Question:** Write a program to swap two numbers without using a third variable.

#include <iostream>

int main() {

int a = 5, b = 10;

a = a + b;

b = a - b;

a = a - b;

std::cout << "a: " << a << ", b: " << b << std::endl;

return 0;

}

**13. GCD of Two Numbers**

**Question:** Write a program to find the GCD of two numbers.

#include <iostream>

int main() {

int a = 56, b = 98;

while (a != b) {

if (a > b)

a -= b;

else

b -= a;

}

std::cout << "GCD: " << a << std::endl;

return 0;

}

**14. LCM of Two Numbers**

**Question:** Write a program to find the LCM of two numbers.

#include <iostream>

int main() {

int a = 15, b = 20, max;

max = (a > b) ? a : b;

while (true) {

if (max % a == 0 && max % b == 0) {

std::cout << "LCM: " << max << std::endl;

break;

}

++max;

}

return 0;

}

**15. Leap Year Check**

**Question:** Write a program to check if a year is a leap year.

#include <iostream>

int main() {

int year = 2020;

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0)

std::cout << "Leap Year" << std::endl;

else

std::cout << "Not Leap Year" << std::endl;

} else

std::cout << "Leap Year" << std::endl;

} else

std::cout << "Not Leap Year" << std::endl;

return 0;

}

**16. Sum of Natural Numbers**

**Question:** Write a program to find the sum of first n natural numbers.

#include <iostream>

int main() {

int n = 10;

int sum = (n \* (n + 1)) / 2;

std::cout << "Sum: " << sum << std::endl;

return 0;

}

**17. Find the Second Largest Number**

**Question:** Write a program to find the second largest number in an array.

#include <iostream>

int main() {

int arr[] = {12, 35, 1, 10, 34, 1};

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

int first = INT\_MIN, second = INT\_MIN;

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

if (arr[i] > first) {

second = first;

first = arr[i];

} else if (arr[i] > second && arr[i] != first) {

second = arr[i];

}

}

std::cout << "Second Largest: " << second << std::endl;

return 0;

}

**18. Count Vowels in a String**

**Question:** Write a program to count the number of vowels in a string.

#include <iostream>

#include <string>

int main() {

std::string str = "Hello World";

int count = 0;

for (char c : str) {

c = tolower(c);

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

count++;

}

}

std::cout << "Number of Vowels: " << count << std::endl;

return 0;

}

**19. Reverse a String**

**Question:** Write a program to reverse a string.

#include <iostream>

#include <string>

int main() {

std::string str = "Hello";

std::string reversed = std::string(str.rbegin(), str.rend());

std::cout << "Reversed: " << reversed << std::endl;

return 0;

}

**20. Remove Duplicates from an Array**

**Question:** Write a program to remove duplicates from an array.

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

std::vector<int> arr = {1, 2, 3, 2, 4, 1, 5};

std::sort(arr.begin(), arr.end());

arr.erase(std::unique(arr.begin(), arr.end()), arr.end());

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;

}

**21. Binary Search**

**Question:** Write a program to implement binary search.

#include <iostream>

#include <vector>

int binarySearch(const std::vector<int>& arr, int l, int r, int x) {

if (r >= l) {

int mid = l + (r - l) / 2;

if (arr[mid] == x)

return mid;

if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);

return binarySearch(arr, mid + 1, r, x);

}

return -1;

}

int main() {

std::vector<int> arr = {2, 3, 4, 10, 40};

int x = 10;

int result = binarySearch(arr, 0, arr.size() - 1, x);

if (result != -1)

std::cout << "Element found at index " << result << std::endl;

else

std::cout << "Element not found" << std::endl;

return 0;

}

**22. Bubble Sort**

**Question:** Write a program to implement bubble sort.

#include <iostream>

#include <vector>

void bubbleSort(std::vector<int>& arr) {

int n = arr.size();

for (int i = 0; i < n - 1; ++i) {

for (int j = 0; j < n - i - 1; ++j) {

if (arr[j] > arr[j + 1]) {

std::swap(arr[j], arr[j + 1]);

}

}

}

}

int main() {

std::vector<int> arr = {64, 34, 25, 12, 22, 11, 90};

bubbleSort(arr);

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;

}

**23. Insertion Sort**

**Question:** Write a program to implement insertion sort.

#include <iostream>

#include <vector>

void insertionSort(std::vector<int>& arr) {

int n = arr.size();

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

int key = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

--j;

}

arr[j + 1] = key;

}

}

int main() {

std::vector<int> arr = {12, 11, 13, 5, 6};

insertionSort(arr);

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;

}

**24. Selection Sort**

**Question:** Write a program to implement selection sort.

#include <iostream>

#include <vector>

void selectionSort(std::vector<int>& arr) {

int n = arr.size();

for (int i = 0; i < n - 1; ++i) {

int minIdx = i;

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

if (arr[j] < arr[minIdx]) {

minIdx = j;

}

}

std::swap(arr[minIdx], arr[i]);

}

}

int main() {

std::vector<int> arr = {64, 25, 12, 22, 11};

selectionSort(arr);

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;

}

**25. Merge Sort**

**Question:** Write a program to implement merge sort.

#include <iostream>

#include <vector>

void merge(std::vector<int>& arr, int l, int m, int r) {

int n1 = m - l + 1;

int n2 = r - m;

std::vector<int> L(n1), R(n2);

for (int i = 0; i < n1; ++i)

L[i] = arr[l + i];

for (int i = 0; i < n2; ++i)

R[i] = arr[m + 1 + i];

int i = 0, j = 0, k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

++i;

} else {

arr[k] = R[j];

++j;

}

++k;

}

while (i < n1) {

arr[k] = L[i];

++i;

++k;

}

while (j < n2) {

arr[k] = R[j];

++j;

++k;

}

}

void mergeSort(std::vector<int>& arr, int l, int r) {

if (l < r) {

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

int main() {

std::vector<int> arr = {12, 11, 13, 5, 6, 7};

mergeSort(arr, 0, arr.size() - 1);

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;

}

**26. Quick Sort**

**Question:** Write a program to implement quick sort.

#include <iostream>

#include <vector>

int partition(std::vector<int>& arr, int low, int high) {

int pivot = arr[high];

int i = (low - 1);

for (int j = low; j <= high - 1; ++j) {

if (arr[j] < pivot) {

++i;

std::swap(arr[i], arr[j]);

}

}

std::swap(arr[i + 1], arr[high]);

return (i + 1);

}

void quickSort(std::vector<int>& arr, int low, int high) {

if (low < high) {

int pi = partition(arr, low, high);

quickSort(arr, low, pi - 1);

quickSort(arr, pi + 1, high);

}

}

int main() {

std::vector<int> arr = {10, 7, 8, 9, 1, 5};

quickSort(arr, 0, arr.size() - 1);

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;

}

**27. Find Maximum in Array**

**Question:** Write a program to find the maximum element in an array.

#include <iostream>

#include <vector>

int main() {

std::vector<int> arr = {10, 324, 45, 90, 9808};

int max = arr[0];

for (int num : arr) {

if (num > max)

max = num;

}

std::cout << "Maximum: " << max << std::endl;

return 0;

}

**28. Find Minimum in Array**

**Question:** Write a program to find the minimum element in an array.

#include <iostream>

#include <vector>

int main() {

std::vector<int> arr = {10, 324, 45, 90, 9808};

int min = arr[0];

for (int num : arr) {

if (num < min)

min = num;

}

std::cout << "Minimum: " << min << std::endl;

return 0;

}

**29. Sum of Array Elements**

**Question:** Write a program to find the sum of all elements in an array.

#include <iostream>

#include <vector>

int main() {

std::vector<int> arr = {1, 2, 3, 4, 5};

int sum = 0;

for (int num : arr) {

sum += num;

}

std::cout << "Sum: " << sum << std::endl;

return 0;

}

**30. Find the Frequency of Each Element in the Array**

**Question:** Write a program to find the frequency of each element in an array.

#include <iostream>

#include <vector>

#include <unordered\_map>

int main() {

std::vector<int> arr = {1, 2, 2, 3, 3, 3, 4};

std::unordered\_map<int, int> freq;

for (int num : arr) {

freq[num]++;

}

for (auto& pair : freq) {

std::cout << pair.first << " occurs " << pair.second << " times" << std::endl;

}

return 0;

}

**31. Transpose of a Matrix**

**Question:** Write a program to find the transpose of a matrix.

#include <iostream>

int main() {

int row = 3, col = 3;

int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

int transpose[3][3];

for (int i = 0; i < row; ++i) {

for (int j = 0; j < col; ++j) {

transpose[j][i] = matrix[i][j];

}

}

std::cout << "Transpose of the matrix:" << std::endl;

for (int i = 0; i < col; ++i) {

for (int j = 0; j < row; ++j) {

std::cout << transpose[i][j] << " ";

}

std::cout << std::endl;

}

return 0;

}

**32. Matrix Multiplication**

**Question:** Write a program to multiply two matrices.

#include <iostream>

int main() {

int row1 = 2, col1 = 3, row2 = 3, col2 = 2;

int matrix1[2][3] = {{1, 2, 3}, {4, 5, 6}};

int matrix2[3][2] = {{7, 8}, {9, 10}, {11, 12}};

int result[2][2] = {0};

if (col1 != row2) {

std::cout << "Matrix multiplication not possible" << std::endl;

return 0;

}

for (int i = 0; i < row1; ++i) {

for (int j = 0; j < col2; ++j) {

for (int k = 0; k < col1; ++k) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

std::cout << "Product of the matrices:" << std::endl;

for (int i = 0; i < row1; ++i) {

for (int j = 0; j < col2; ++j) {

std::cout << result[i][j] << " ";

}

std::cout << std::endl;

}

return 0;

}

**33. Find the Length of a String**

**Question:** Write a program to find the length of a string.

#include <iostream>

#include <cstring>

int main() {

char str[] = "Hello World";

int length = std::strlen(str);

std::cout << "Length: " << length << std::endl;

return 0;

}

**34. Count Words in a String**

**Question:** Write a program to count the number of words in a string.

#include <iostream>

#include <string>

#include <sstream>

int main() {

std::string str = "Hello world, welcome to C++ programming";

std::stringstream ss(str);

std::string word;

int count = 0;

while (ss >> word) {

count++;

}

std::cout << "Number of Words: " << count << std::endl;

return 0;

}

**35. Reverse Words in a String**

**Question:** Write a program to reverse the words in a string.

#include <iostream>

#include <string>

#include <sstream>

#include <vector>

int main() {

std::string str = "Hello world, welcome to C++ programming";

std::stringstream ss(str);

std::string word;

std::vector<std::string> words;

while (ss >> word) {

words.push\_back(word);

}

std::reverse(words.begin(), words.end());

for (const std::string& w : words) {

std::cout << w << " ";

}

std::cout << std::endl;

return 0;

}

**36. Find Substring**

**Question:** Write a program to find a substring in a string.

#include <iostream>

#include <string>

int main() {

std::string str = "Hello world, welcome to C++ programming";

std::string substr = "welcome";

size\_t found = str.find(substr);

if (found != std::string::npos) {

std::cout << "Substring found at index " << found << std::endl;

} else {

std::cout << "Substring not found" << std::endl;

}

return 0;

}

**37. Replace Substring**

**Question:** Write a program to replace a substring in a string.

#include <iostream>

#include <string>

int main() {

std::string str = "Hello world, welcome to C++ programming";

std::string oldSubstr = "world";

std::string newSubstr = "everyone";

size\_t found = str.find(oldSubstr);

if (found != std::string::npos) {

str.replace(found, oldSubstr.length(), newSubstr);

}

std::cout << str << std::endl;

return 0;

}

**38. Check for Anagram**

**Question:** Write a program to check if two strings are anagrams.

#include <iostream>

#include <algorithm>

#include <string>

bool areAnagrams(const std::string& str1, const std::string& str2) {

if (str1.length() != str2.length())

return false;

std::string s1 = str1, s2 = str2;

std::sort(s1.begin(), s1.end());

std::sort(s2.begin(), s2.end());

return s1 == s2;

}

int main() {

std::string str1 = "listen";

std::string str2 = "silent";

if (areAnagrams(str1, str2))

std::cout << "Anagrams" << std::endl;

else

std::cout << "Not Anagrams" << std::endl;

return 0;

}

**39. Count Character Frequency**

**Question:** Write a program to count the frequency of each character in a string.

#include <iostream>

#include <string>

#include <unordered\_map>

int main() {

std::string str = "Hello world";

std::unordered\_map<char, int> freq;

for (char c : str) {

freq[c]++;

}

for (auto& pair : freq) {

std::cout << pair.first << ": " << pair.second << std::endl;

}

return 0;

}

**40. Remove Whitespaces**

**Question:** Write a program to remove whitespaces from a string.

#include <iostream>

#include <string>

int main() {

std::string str = " Hello World ";

str.erase(remove(str.begin(), str.end(), ' '), str.end());

std::cout << str << std::endl;

return 0;

}

**41. Check if String is Palindrome**

**Question:** Write a program to check if a string is a palindrome.

#include <iostream>

#include <string>

int main() {

std::string str = "madam";

std::string reversed = std::string(str.rbegin(), str.rend());

if (str == reversed)

std::cout << "Palindrome" << std::endl;

else

std::cout << "Not Palindrome" << std::endl;

return 0;

}

**42. Check Armstrong Number for n Digits**

**Question:** Write a program to check if a number is an Armstrong number for n digits.

#include <iostream>

#include <cmath>

int main() {

int n = 9474;

int sum = 0, temp = n, digits = 0;

while (temp != 0) {

digits++;

temp /= 10;

}

temp = n;

while (temp != 0) {

int digit = temp % 10;

sum += std::pow(digit, digits);

temp /= 10;

}

if (sum == n)

std::cout << "Armstrong" << std::endl;

else

std::cout << "Not Armstrong" << std::endl;

return 0;

}

**43. Check if Number is Prime**

**Question:** Write a program to check if a number is prime.

#include <iostream>

bool isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i \* i <= n; ++i) {

if (n % i == 0) return false;

}

return true;

}

int main() {

int n = 29;

if (isPrime(n))

std::cout << "Prime" << std::endl;

else

std::cout << "Not Prime" << std::endl;

return 0;

}

**44. Factorial of a Number**

**Question:** Write a program to find the factorial of a number.

#include <iostream>

int factorial(int n) {

return (n <= 1) ? 1 : n \* factorial(n - 1);

}

int main() {

int n = 5;

std::cout << "Factorial: " << factorial(n) << std::endl;

return 0;

}

**45. Fibonacci Series**

**Question:** Write a program to generate the Fibonacci series up to n terms.

#include <iostream>

void fibonacci(int n) {

int t1 = 0, t2 = 1, nextTerm;

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

std::cout << t1 << " ";

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

}

std::cout << std::endl;

}

int main() {

int n = 10;

fibonacci(n);

return 0;

}

**46. Sum of Digits**

**Question:** Write a program to find the sum of the digits of a number.

#include <iostream>

int main() {

int n = 1234, sum = 0;

while (n != 0) {

sum += n % 10;

n /= 10;

}

std::cout << "Sum of digits: " << sum << std::endl;

return 0;

}

**47. Reverse a Number**

**Question:** Write a program to reverse a number.

#include <iostream>

int main() {

int n = 1234, reversed = 0;

while (n != 0) {

int digit = n % 10;

reversed = reversed \* 10 + digit;

n /= 10;

}

std::cout << "Reversed number: " << reversed << std::endl;

return 0;

}

**48. Check if a Number is a Palindrome**

**Question:** Write a program to check if a number is a palindrome.

#include <iostream>

int main() {

int n = 121, original = n, reversed = 0;

while (n != 0) {

int digit = n % 10;

reversed = reversed \* 10 + digit;

n /= 10;

}

if (original == reversed)

std::cout << "Palindrome" << std::endl;

else

std::cout << "Not Palindrome" << std::endl;

return 0;

}

**49. Find GCD of Two Numbers**

**Question:** Write a program to find the GCD of two numbers.

#include <iostream>

int gcd(int a, int b) {

return (b == 0) ? a : gcd(b, a % b);

}

int main() {

int a = 56, b = 98;

std::cout << "GCD: " << gcd(a, b) << std::endl;

return 0;

}

**50. Find LCM of Two Numbers**

**Question:** Write a program to find the LCM of two numbers.

#include <iostream>

int gcd(int a, int b) {

return (b == 0) ? a : gcd(b, a % b);

}

int lcm(int a, int b) {

return (a / gcd(a, b)) \* b;

}

int main() {

int a = 12, b = 15;

std::cout << "LCM: " << lcm(a, b) << std::endl;

return 0;

}

**51. Find the Power of a Number**

**Question:** Write a program to find the power of a number.

#include <iostream>

int main() {

int base = 2, exponent = 3;

int result = 1;

for (int i = 0; i < exponent; ++i) {

result \*= base;

}

std::cout << "Power: " << result << std::endl;

return 0;

}

**52. Find nCr**

**Question:** Write a program to find the binomial coefficient nCr.

#include <iostream>

int factorial(int n) {

return (n <= 1) ? 1 : n \* factorial(n - 1);

}

int nCr(int n, int r) {

return factorial(n) / (factorial(r) \* factorial(n - r));

}

int main() {

int n = 5, r = 2;

std::cout << "nCr: " << nCr(n, r) << std::endl;

return 0;

}

**53. Find nPr**

**Question:** Write a program to find the number of permutations nPr.

#include <iostream>

int factorial(int n) {

return (n <= 1) ? 1 : n \* factorial(n - 1);

}

int nPr(int n, int r) {

return factorial(n) / factorial(n - r);

}

int main() {

int n = 5, r = 2;

std::cout << "nPr: " << nPr(n, r) << std::endl;

return 0;

}

**54. Check if Year is Leap Year**

**Question:** Write a program to check if a year is a leap year.

#include <iostream>

int main() {

int year = 2024;

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))

std::cout << "Leap Year" << std::endl;

else

std::cout << "Not Leap Year" << std::endl;

return 0;

}

**55. Convert Decimal to Binary**

**Question:** Write a program to convert a decimal number to binary.

#include <iostream>

#include <stack>

int main() {

int decimal = 10;

std::stack<int> binary;

while (decimal > 0) {

binary.push(decimal % 2);

decimal /= 2;

}

std::cout << "Binary: ";

while (!binary.empty()) {

std::cout << binary.top();

binary.pop();

}

std::cout << std::endl;

return 0;

}

**56. Convert Binary to Decimal**

**Question:** Write a program to convert a binary number to decimal.

#include <iostream>

#include <cmath>

int main() {

int binary = 1010;

int decimal = 0, base = 1, temp = binary;

while (temp > 0) {

int lastDigit = temp % 10;

temp /= 10;

decimal += lastDigit \* base;

base \*= 2;

}

std::cout << "Decimal: " << decimal << std::endl;

return 0;

}

**57. Convert Decimal to Hexadecimal**

**Question:** Write a program to convert a decimal number to hexadecimal.

#include <iostream>

int main() {

int decimal = 2545;

char hex[100];

int i = 0;

while (decimal != 0) {

int temp = decimal % 16;

if (temp < 10)

hex[i] = temp + 48;

else

hex[i] = temp + 55;

decimal /= 16;

i++;

}

std::cout << "Hexadecimal: ";

for (int j = i - 1; j >= 0; j--)

std::cout << hex[j];

std::cout << std::endl;

return 0;

}

**58. Convert Hexadecimal to Decimal**

**Question:** Write a program to convert a hexadecimal number to decimal.

#include <iostream>

#include <cstring>

#include <cmath>

int main() {

char hex[100];

std::cout << "Enter a hexadecimal number: ";

std::cin >> hex;

int length = std::strlen(hex);

int base = 1, decimal = 0;

for (int i = length - 1; i >= 0; i--) {

if (hex[i] >= '0' && hex[i] <= '9') {

decimal += (hex[i] - 48) \* base;

base \*= 16;

} else if (hex[i] >= 'A' && hex[i] <= 'F') {

decimal += (hex[i] - 55) \* base;

base \*= 16;

}

}

std::cout << "Decimal: " << decimal << std::endl;

return 0;

}

**59. Generate Random Numbers**

**Question:** Write a program to generate random numbers.

#include <iostream>

#include <cstdlib>

#include <ctime>

int main() {

std::srand(std::time(0));

for (int i = 0; i < 10; ++i) {

std::cout << std::rand() % 100 << " ";

}

std::cout << std::endl;

return 0;

}

**60. Find the Sum of the Series**

**Question:** Write a program to find the sum of the series 1 + 1/2 + 1/3 + ... + 1/n.

#include <iostream>

int main() {

int n = 10;

double sum = 0.0;

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

sum += 1.0 / i;

}

std::cout << "Sum: " << sum << std::endl;

return 0;

}

Top of Form

Bottom of Form