Run | Debug | Stop | Share | Save | { } Beautify

main.cpp

```cpp
#include <iostream>
using namespace std;

bool isPrime(int num, int i) {
    if (num <= 2)
        return (num == 2) ? true : false;
    if (num % i == 0)
        return false;
    if (i * i > num)
        return true;
    return isPrime(num, i + 1);
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    if (isPrime(num, 2))
        cout << num << " is a prime number.";
    else
        cout << num << " is not a prime number.";
    return 0;
}
```

input

```
Enter a number: 13
13 is a prime number.

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | { } Beautify

main.cpp

```cpp
#include <iostream>
#include <string.h>
using namespace std;

void reverseStr(char* str, int start, int end) {
    if (start >= end)
        return;
    swap(str[start], str[end]);
    reverseStr(str, start + 1, end - 1);
}

int main() {
    char str[100];
    cout << "Enter a string: ";
    cin >> str;
    reverseStr(str, 0, strlen(str) - 1);
    cout << "Reversed string: " << str;
    return 0;
}
```

input

```
Enter a string: premalu
Reversed string: ulamerp

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | {} Beautify

main.cpp

```
    if (num <= 2)
        return (num == 2) ? true : false;
    if (num % i == 0)
        return false;
    if (i * i > num)
        return true;
    return isPrime(num, i + 1);
}

void printPrimes(int n, int current) {
    if (current > n)
        return;
    if (isPrime(current, 2))
        cout << current << " ";
    printPrimes(n, current + 1);
}

int main() {
    int n;
    cout << "Enter the limit: ";
    cin >> n;
    printPrimes(n, 2);
    return 0;
}
```

input

```
Enter the limit: 20
2 3 5 7 11 13 17 19

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | { } Beautify

main.cpp

```cpp
#include <iostream>
using namespace std;

void strCopy(char* source, char* dest, int index) {
    dest[index] = source[index];
    if (source[index] == '\0')
        return;
    strCopy(source, dest, index + 1);
}

int main() {
    char source[100], dest[100];
    cout << "Enter a string: ";
    cin >> source;
    strCopy(source, dest, 0);
    cout << "Copied string: " << dest;
    return 0;
}
```

input

```
Enter a string: happy
Copied string: happy

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Saved | {} Beautify

main.cpp

```cpp
#include <iostream>
using namespace std;

int factorial(int n) {
    if (n <= 1)
        return 1;
    return n * factorial(n - 1);
}

int main() {
    int n;
    cout << "Enter a number: ";
    cin >> n;
    cout << "Factorial: " << factorial(n);
    return 0;
}

```

```
Enter a number: 5
Factorial: 120

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | {} Beautify

main.cpp

```cpp
#include <iostream>
using namespace std;

int findLargest(int arr[], int n) {
    if (n == 1)
        return arr[0];
    return max(arr[n-1], findLargest(arr, n-1));
}

int main() {
    int n;
    cout << "Enter number of elements: ";
    cin >> n;
    int arr[n];
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];
    cout << "Largest element: " << findLargest(arr, n);
    return 0;
}
```

input

```
Enter number of elements: 5
Enter elements: 1  5 25 75 99
Largest element: 99

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | {} Beautify

main.cpp

```cpp
#include <iostream>
using namespace std;

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

int main() {
    int a, b;
    cout << "Enter two numbers: ";
    cin >> a >> b;
    cout << "GCD: " << gcd(a, b);
    return 0;
}

```

input

```
Enter two numbers: 12
24
GCD: 12

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | { } Beautify

main.cpp

```cpp
#include <iostream>
using namespace std;

int fib(int n) {
    if (n <= 1)
        return n;
    return fib(n - 1) + fib(n - 2);
}

int main() {
    int n;
    cout << "Enter the number of terms: ";
    cin >> n;
    for (int i = 0; i < n; i++)
        cout << fib(i) << " ";
    return 0;
}
```

input

```
Enter the number of terms: 5
0 1 1 2 3

...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Saved | { } Beautify

main.cpp

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

int power(int num, int p) {
    if (p == 0) return 1;
    return num * power(num, p - 1);
}
int sumOfPoweredDigits(int num, int len) {
    if (num == 0) return 0;
    return power(num % 10, len) + sumOfPoweredDigits(num / 10, len);
}


bool isArmstrong(int num) {
    int len = to_string(num).length();
    int sum = sumOfPoweredDigits(num, len);
    return sum == num;
}

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

    if (isArmstrong(num))
        cout << num << " is an Armstrong number." << endl;
```

input

```
Enter a number: 153
153 is an Armstrong number.


...Program finished with exit code 0
Press ENTER to exit console.
```

Run | Debug | Stop | Share | Save | { } Beautify

main.cpp

```cpp
#include <iostream>
#include<string.h>
using namespace std;

bool isPalindrome(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 >> str;
    if (isPalindrome(str, 0, strlen(str) - 1))
        cout << str << " is a palindrome.";
    else
        cout << str << " is not a palindrome.";
    return 0;
}
```

input

```
Enter a string: madam
madam is a palindrome.

...Program finished with exit code 0
Press ENTER to exit console.
```