

Armstrong number

```
#include <iostream> using namespace std;
int main() {
    int num, original, remainder, result = 0;
    cout << "Enter a number: ";
    cin >> num;
    original = num; // Store the original number
    while (num != 0) {
        remainder = num % 10;    // Get last digit
        result += remainder * remainder * remainder; // Cube and add
        num /= 10;              // Remove last digit
    }
    if (result == original)
        cout << original << " is an Armstrong number." << endl;
    else
        cout << original << " is NOT an Armstrong number." << endl;
    return 0;
}
```

Prime Number upto n

```
#include <iostream> using namespace std;
bool isPrime(int num) {
    if (num <= 1) return false; // 0 and 1 are not prime numbers
    if (num == 2) return true;
    if (num % 2 == 0) return false;
    for(int i=3, i*i <= num; i+=2){    if (num % i==0) return false;
    }
    return true; // num is prime
}
int main() { int n; cout << "enter n"; cin >> n;
    cout << "Prime numbers up to " << n << " are : \n";
    for (int i = 2; i <= n; i++) {
        if (isPrime(i)) {
            cout << i << " ";
        }
    }
    cout << endl;
    return 0;
}
```

Multiplication and Addition: #include <iostream> using namespace std;

```
int main() {
    int p, q;
    cout << "Enter the value of p: "; cin >> p;
    cout << "Enter the value of q: "; cin >> q;
    cout << "The sum of p and q is: " << p + q << endl;
    cout << "The product of p and q is: " << p * q << endl;
    return 0; }
```

Rev String: #include <iostream>#include <string> using namespace std;

```
int main()
{ string str;
  cout << "Enter a string: ";
  cin >> str;
  int i=0; int j = str.length()-1;
  while (i < j){
    char temp;
    temp = str[i];
    str[i] = str[j];
    str[j] = temp;
    i++;
    j--;
  }
  cout << "Rev String is:" << str;
}
```

Fibonacci series #include <iostream> using namespace std;

```
int main() { int n, t1 = 0, t2 = 1, nextTerm;
  cout << "Enter number of terms: "; cin >> n;
  cout << "Fibonacci Series: ";
  for (int i = 1; i <= n; i++) {
    cout << t1 << " "; // Print current term
    nextTerm = t1 + t2; // Calculate next term
    t1 = t2; // Move forward
    t2 = nextTerm;
  } return 0; }
```

Matrix and Transpose:#include <iostream> using namespace std;

```
int main(){
  int matrix[3][3]={1,2,3,4,5,6,7,8,9}; int transpose[3][3];
  cout<<"The Matrix is:"<< endl;
  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++)
      {cout<<"The Matrix is:"<< endl;
        cout<<matrix[i][j]<<"\t";
      }cout<<endl;
    } cout <<"======"<<endl;
  for (int i = 0; i < 3; i++)
  { for (int j = 0; j < 3; j++)
    { transpose[j][i]=matrix[j][i];
    }
  }; for (int i = 0; i < 3; i++)
  { for (int j = 0; j < 3; j++)
    {
      cout<< transpose[j][i]<<"\t";
    } cout<<endl;
  }return 0;
}
```

Second Largest: #include <iostream> using namespace std;

```
int main(){
    int n;
    cout <<"Enter size of Array: ";
    cin>>n;
    if (n<2)
    {
        cout<<"Invalid size of array:"<< endl;
        return 1;
    }
    int elements[n];
    cout<<"Enter "<<n<<" elements to find the second largest number: \n";
    for (int i = 0; i < n; ++i)
    {
        cin>>elements[i];
    }
    int largest , secondLargest; // Initialize 2 elements to compare largest and secondLargest
    if (elements[0]>elements[1])
    {
        largest = elements[0];
        secondLargest = elements[1];
    }
    for (int i = 2; i < n; i++)//Process to run loop for 'n' number of elements
    {
        if (elements[i]>largest){
            secondLargest=largest;
            largest=elements[i];
        } else if (elements[i] > secondLargest && elements[i] != largest)
        {
            secondLargest = elements[i];
        }
    }
    if (largest == secondLargest)
    {
        cout<< "Error: All elements same /No second largest element found"<< endl;
    }
    else {
        cout << "The second largest number is: "<<secondLargest<<endl;
    }

    return 0;
}
```

Multiply Matrix: #include <iostream> using namespace std;

```
int main() {
    int a[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    int b[3][3] = {9, 8, 7, 6, 5, 4, 3, 2, 1};
    int result[3][3] = {0}; // Initialize all elements to 0
    for(int i = 0; i < 3; i++) { // Multiply matrices
        for(int j = 0; j < 3; j++) {
            for(int k = 0; k < 3; k++) {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    cout << "Product of the matrices:\n"; // Display result
    for(int i = 0; i < 3; i++) {
        for(int j = 0; j < 3; j++) {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Factorial : #include <iostream> using namespace std;

```
int main() {
    int n;
    unsigned long long factorial = 1; // Use long long for large values
    cout << "Enter a positive integer: ";
    cin >> n;
    if (n < 0) {
        cout << "Factorial of a negative number doesn't exist." << endl;
    } else {
        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }
        cout << "Factorial of " << n << " = " << factorial << endl;
    }
    return 0;
}
```

Palindrome String: #include <iostream>

```
#include <string>
using namespace std;
int main() { string str, rev;
    cout << "Enter a string: "; cin >> str;
    rev = string(str.rbegin(), str.rend()); // reverse using iterators
    if (str == rev)
        cout << str << " is a palindrome string." << endl;
    else
        cout << str << " is not a palindrome string." << endl;
    return 0; }
```

Palindrome Number:

```
int num, reversed = 0, original, remainder; cout << "Enter a number: ";
cin >> num; original = num; while (num != 0) { remainder = num % 10;
reversed = reversed * 10 + remainder; num /= 10;}
```