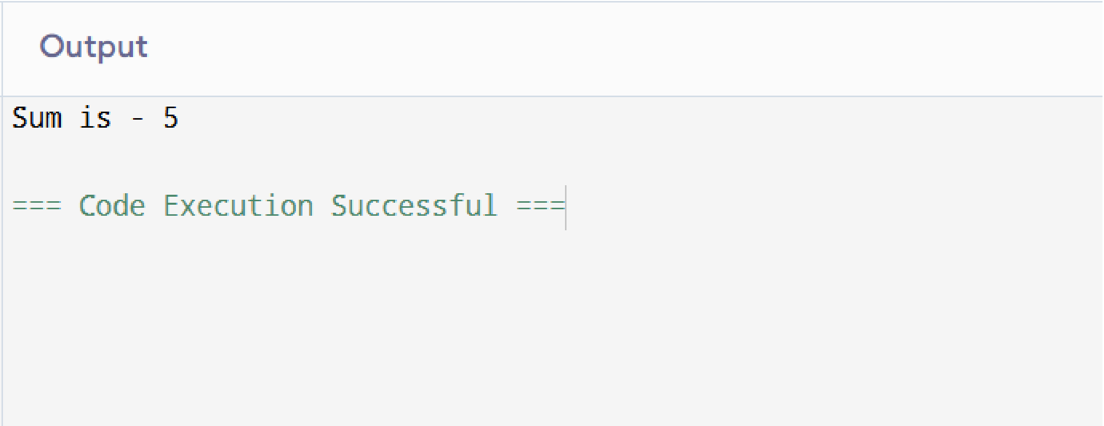
Day - 3

Sum of Two Numbers

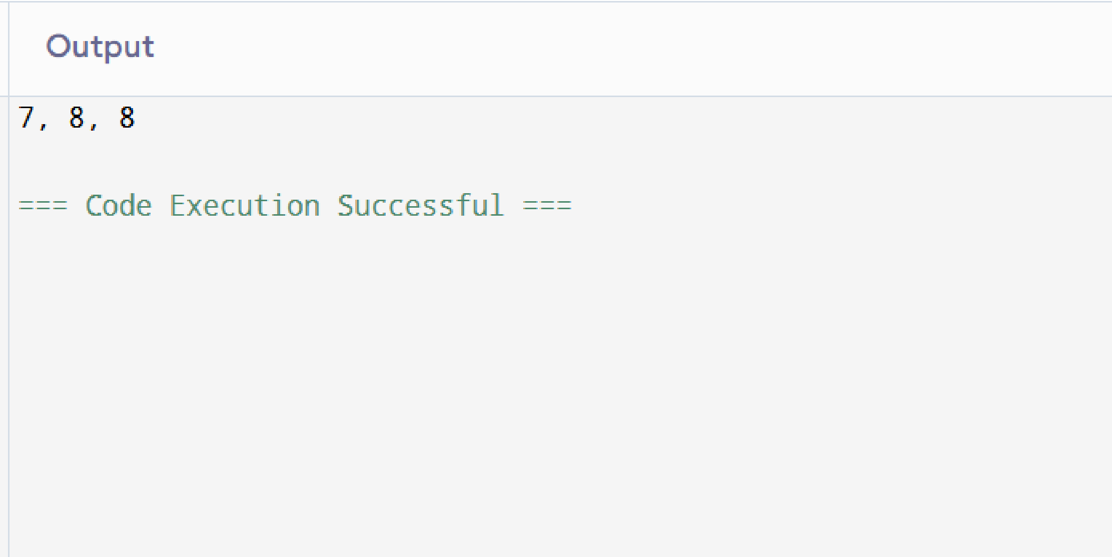
```
#include <iostream>
using namespace std;
int sum(int a , int b){
 return a+b;
}
int main() {
 int ans= sum(2,3);
 cout<<"Sum is - "<<ans;
 return 0;
}
```



```
Sum is - 5

=== Code Execution Successful ===
```

Add Corresponding Elements of Arrays

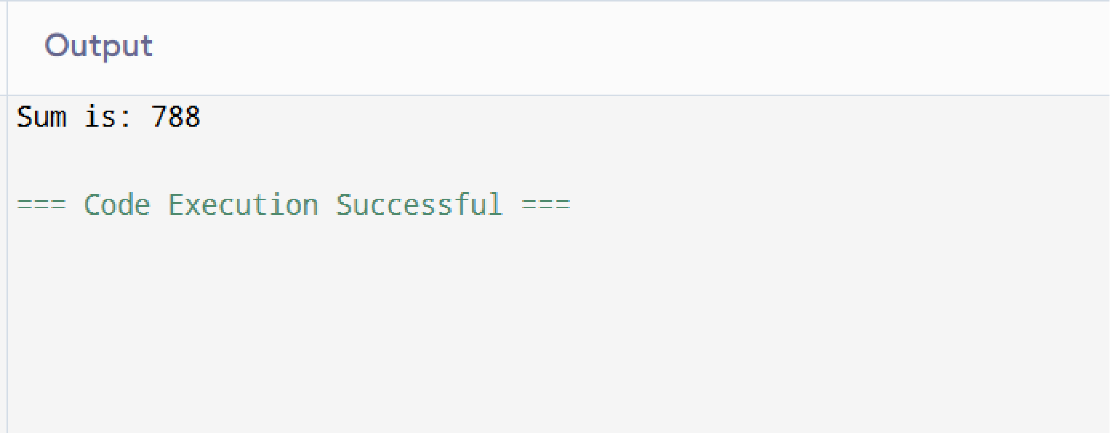
```
#include <iostream>
using namespace std;
int main() {
 const int size = 3;
 int a[size] = {2, 4, 3};
 int b[size] = {5, 4, 5};
 int result[size];
 // Adding corresponding elements of the two arrays
 for (int i = 0; i < size; ++i) {
 result[i] = a[i] + b[i];
 }
 // Printing the result array with proper formatting
 for (int i = 0; i < size; ++i) {
 cout << result[i];
 if (i != size - 1) { // Add a comma and space except for the last element
 cout << ", ";
 }
 }
 return 0;
}
```



```
7, 8, 8

=== Code Execution Successful ===
```

Add Two Numbers Formed by Arrays

```
#include <iostream>
using namespace std;
int main() {
 const int size = 3;
 int a[size] = {2, 4, 3};
 int b[size] = {5, 4, 5};
 int num1 = 0, num2 = 0;
 for (int i = 0; i < size; ++i) {
 num1 = num1 * 10 + a[i];
 num2 = num2 * 10 + b[i];
 }
 int sum = num1 + num2;
 cout << "Sum is: " << sum;
 return 0;
}
```

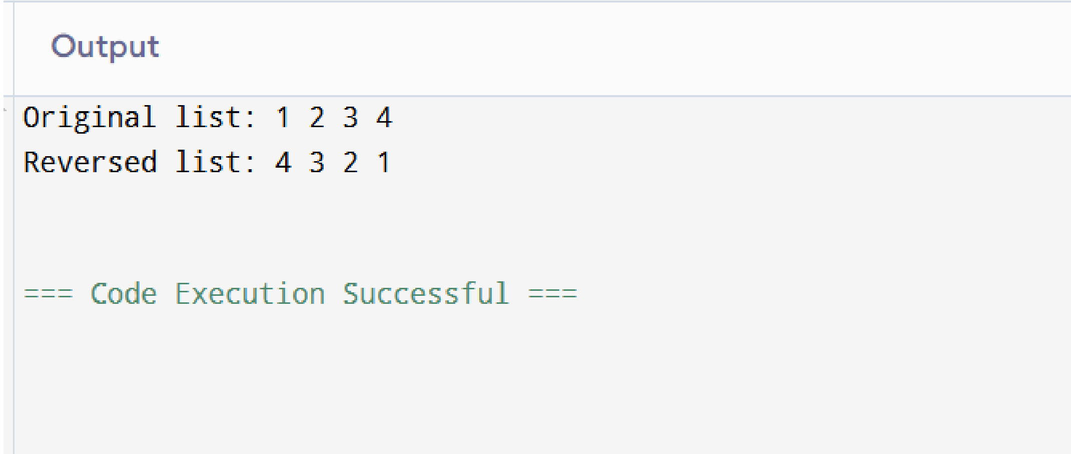


```
Sum is: 788

=== Code Execution Successful ===
```

Reverse a Linked List

```
#include <iostream>
using namespace std;
struct Node {
 int data;
 Node* next;
};
Node* reverseLinkedList(Node* head) {
 Node* prev = nullptr;
 Node* current = head;
 Node* next = nullptr;
 while (current != nullptr) {
 next = current->next;
 current->next = prev;
 prev = current;
 current = next;
 }
 return prev;
}
void printList(Node* head) {
 Node* temp = head;
 while (temp != nullptr) {
 cout << temp->data << " ";
 temp = temp->next;
 }
 cout << endl;
}
```

```
void push(Node** head, int data) {
 Node* newNode = new Node();
 newNode->data = data;
 newNode->next = *head;
 *head = newNode;
}
int main() {
 Node* head = nullptr;
 push(&head, 4);
 push(&head, 3);
 push(&head, 2);
 push(&head, 1);
 cout << "Original list: ";
 printList(head);
 head = reverseLinkedList(head);
 cout << "Reversed list: ";
 printList(head);
 return 0;
}
```

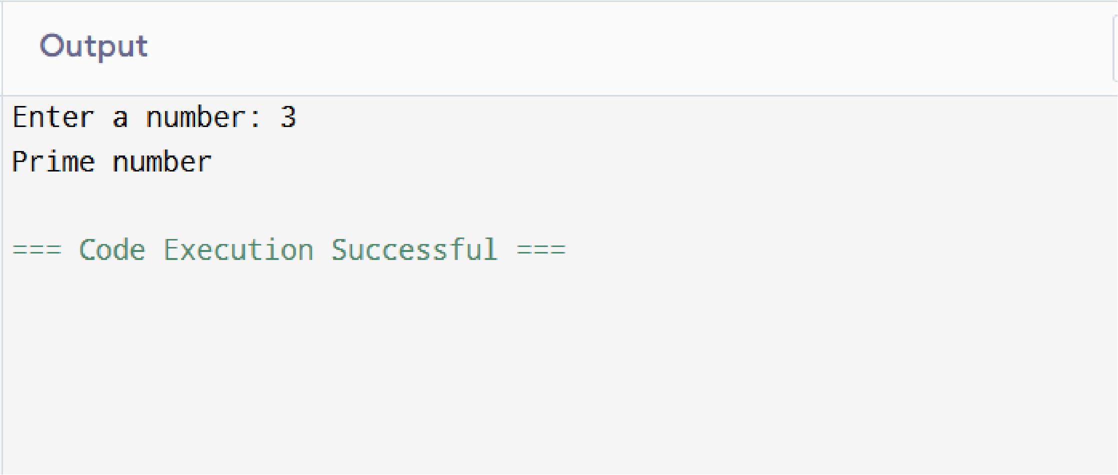


Output

```
Original list: 1 2 3 4
Reversed list: 4 3 2 1


=== Code Execution Successful ===
```

Check Prime Number

```
#include<iostream>
using namespace std;
void prime(int n) {
 int count = 0;
 for(int i = 1; i <= n; i++) {
 if(n % i == 0) {
 count++;
 }
 }
 if(count == 2) {
 cout << "Prime number ";
 } else {
 cout << "Not a prime number ";
 }
}
int main() {
 int n;
 cout << "Enter a number: ";
 cin >> n;
 prime(n);
 return 0;
}
```

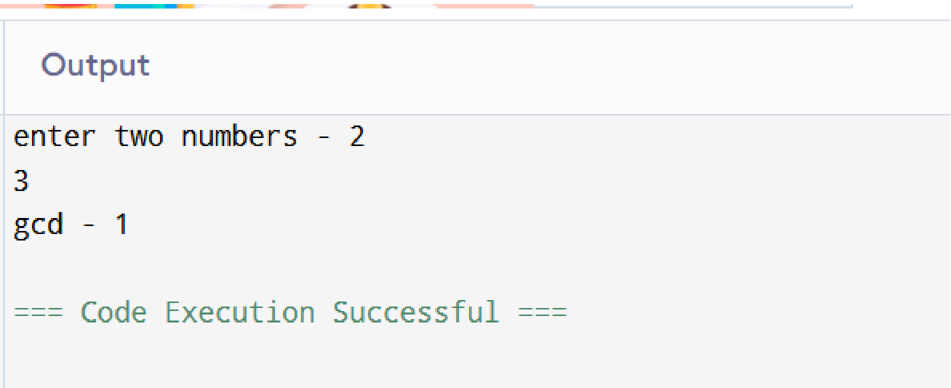


```
Enter a number: 3
Prime number

=== Code Execution Successful ===
```

Find GCD of Two Numbers

```
#include<iostream>
using namespace std;
void gcd( int n1,int n2){
 while(n1!=n2){
 if(n1>n2){
 n1=n1-n2;}
 else{
 n2=n2-n1;
 }
 }
 cout<< "gcd - "<<n1;
}
```

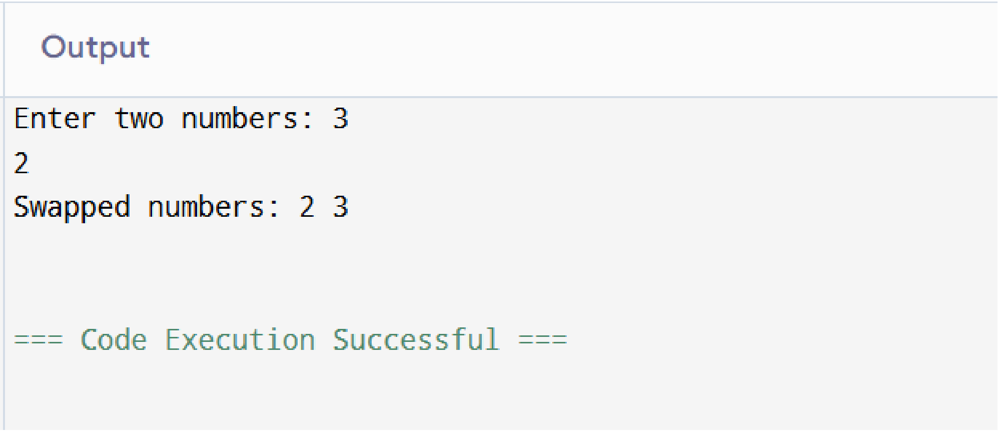
```
int main(){
 int n1,n2;
 cout<<"enter two numbers - ";
 cin>>n1>>n2;
 gcd(n1,n2); return 0; }
```



```
enter two numbers - 2
3
gcd - 1

=== Code Execution Successful ===
```

Swap Two Numbers

```
#include<iostream>
using namespace std;
void swap(int &a, int &b) {
 int c = a;
 a = b;
 b = c;
 cout << "Swapped numbers: " << a << " " << b << endl;
}
int main() {
 int a, b;
 cout << "Enter two numbers: ";
 cin >> a >> b;
 swap(a, b);
 return 0;
}
```

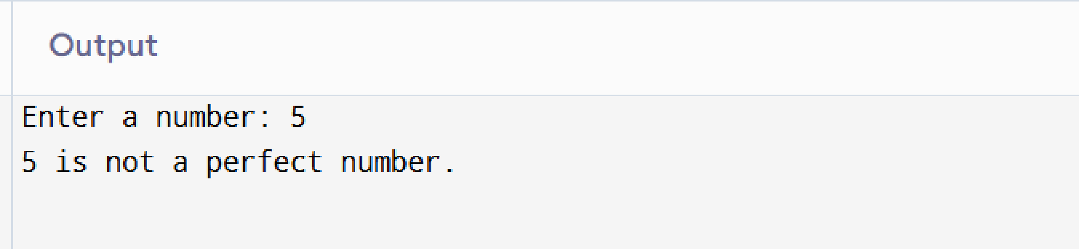


Output

```
Enter two numbers: 3
2
Swapped numbers: 2 3


=== Code Execution Successful ===
```

Check perfect square

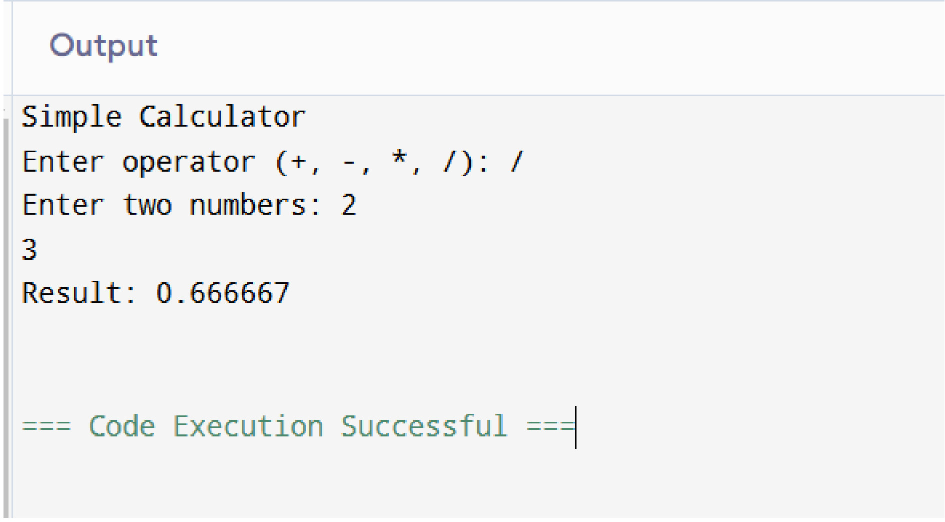
```
#include <iostream>
using namespace std;
bool isPerfectNumber(int num) {
 if (num <= 1) return false;
 int sum = 0;
 for (int i = 1; i <= num / 2; i++) {
 if (num % i == 0) {
 sum += i;
 }
 }

 return sum == num;
}
int main() {
 int num;
 cout << "Enter a number: ";
 cin >> num;
 if (isPerfectNumber(num)) {
 cout << num << " is a perfect number." << endl;
 } else {
 cout << num << " is not a perfect number." << endl;
 }
 return 0;
}
```



```
Enter a number: 5
5 is not a perfect number.
```

Program of simple calculator

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

int main() {
    double num1, num2;
    char op;

    // Display the menu to the user
    cout << "Simple Calculator" << endl;
    cout << "Enter operator (+, -, *, /): ";
    cin >> op;

    // Ask the user to enter two numbers
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

    // Perform the calculation based on the operator entered
    switch(op) {
        case '+':
            cout << "Result: " << num1 + num2 << endl;
            break;
        case '-':
            cout << "Result: " << num1 - num2 << endl;
            break;
        case '*':
            cout << "Result: " << num1 * num2 << endl;
            break;
        case '/':
            // Check for division by zero
            if(num2 != 0)
                cout << "Result: " << num1 / num2 << endl;
            else
                cout << "Error! Division by zero." << endl;
            break;
        default:
            cout << "Invalid operator!" << endl;
    }

    return 0;
}
```

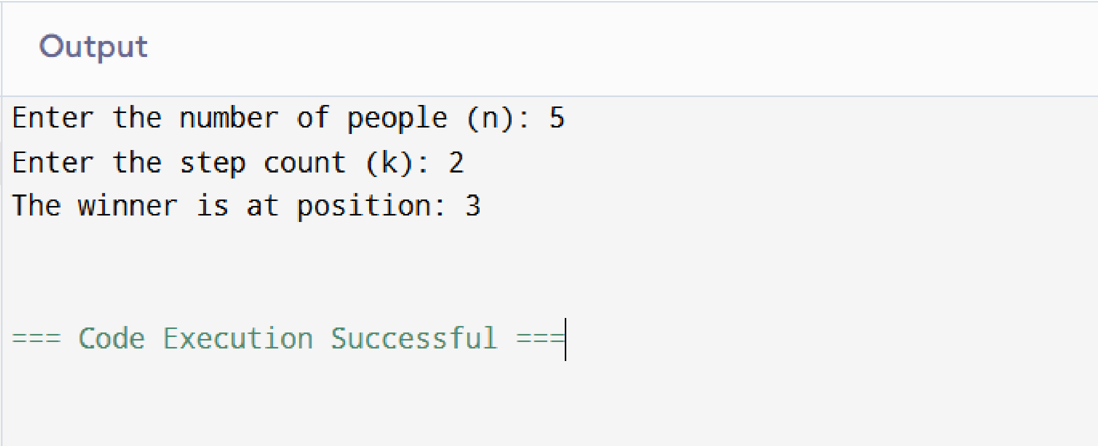


```
Simple Calculator
Enter operator (+, -, *, /): /
Enter two numbers: 2
3
Result: 0.666667


=== Code Execution Successful ===
```

Find the winner of circular game

```
#include <iostream>
using namespace std;
int josephus(int n, int k) {
   if (n == 1) {
      return 0;
   } else {
      return (josephus(n - 1, k) + k) % n;  // Recurrence relation
   }
}
int main() {
   int n, k;
   cout << "Enter the number of people (n): ";
   cin >> n;

   cout << "Enter the step count (k): ";
   cin >> k;
   int winner = josephus(n, k) + 1;
```

```
    cout << "The winner is at position: " << winner << endl;


    return 0;
}
```



```
Enter the number of people (n): 5
Enter the step count (k): 2
The winner is at position: 3


=== Code Execution Successful ===
```