A:

#include <iostream>

#include <vector>

using namespace std;

void bubbleSort(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]) {

// Swap arr[j] and arr[j+1]

int temp = arr[j];

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

arr[j+1] = temp;

}

}

}

}

int main() {

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

bubbleSort(arr);

for (int x : arr) {

cout << x << " ";

}

cout << endl;

return 0;

}

B:

#include <iostream>

#include <vector>

using namespace std;

void bubbleSort(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]) {

int temp = arr[j]

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

arr[j+1] = temp;

}

}

}

}

int main() {

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

bubbleSort(arr);

for (int x : arr) {

cout << x << " ";

}

cout << endl;

return 0;

}

C:

#include <iostream>

#include <vector>

using namespace std;

void bubbleSort(vector<int>& arr) {

int n = arr.size();

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

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

// Bug: Should compare arr[j] > arr[j+1] but incorrectly compares arr[j] > arr[i]

if (arr[j] > arr[i]) {

int temp = arr[j];

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

arr[j+1] = temp;

}

}

}

}

int main() {

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

bubbleSort(arr);

for (int x : arr) {

cout << x << " ";

}

cout << endl;

return 0;

}

D:

#include <iostream>

#include <vector>

using namespace std;

void bubbleSort(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]) {

int temp = arr[j];

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

arr[j+1] = temp;

}

}

}

}

int main() {

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

bubbleSort(arr);

for (int x : arr) {

cout << x << " ";

}

cout << endl;

return 0;

}

E:

using System;

public class Fibonacci

{

public static int FibonacciNumber(int n)

{

if (n < 0)

{

throw new ArgumentException("Input must be non-negative");

}

if (n == 0) return 0;

if (n == 1) return 1;

int a = 0, b = 1;

int c = 0;

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

{

c = a + b;

a = b;

b = c;

}

return fib;

}

public static void Main()

{

Console.WriteLine("Fibonacci of 10 is: " + FibonacciNumber(10));

}

}

F:

using System;

public class Fibonacci

{

public static int FibonacciNumber(int n)

{

if (n < 0)

{

throw new ArgumentException("Input must be non-negative");

}

if (n == 0) return 0;

if (n == 1) return 1;

int a = 0, b = 1;

int c = 0;

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

{

c = a + b;

a = b;

b = c;

}

return c;

}

public static void Main()

{

Console.WriteLine("Fibonacci of 10 is: " + FibonacciNumber(10));

}

}

G:

using System;

public class Fibonacci

{

public static int FibonacciNumber(int n)

{

if (n < 0)

{

throw new ArgumentException("Input must be non-negative")

}

if (n == 0) return 0;

if (n == 1) return 1;

int a = 0, b = 1;

int c = 0;

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

{

c = a + b;

a = b;

b = c;

}

return c;

}

public static void Main()

{

Console.WriteLine("Fibonacci of 10 is: " + FibonacciNumber(10));

}

}

H:

using System;

public class Fibonacci

{

public static int FibonacciNumber(int n)

{

if (n < 0)

{

throw new ArgumentException("Input must be non-negative");

}

if (n == 0) return 0;

if (n == 1) return 1;

int a = 0, b = 1;

int c = 0;

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

{

c = a + b;

a = b;

b = c;

}

return c;

}

public static void Main()

{

Console.WriteLine("Fibonacci of 10 is: " + FibonacciNumber(10)); // Expected output: 55

}

}