// This program is a linear search function.

/\*#include <iostream>

using namespace std;

int main()

{

int searchList(int list[], int numElems, int value);

int value;

int numElems = 5;

int list[5] = { 12, 15, 0, 2, 6 };

value = searchList(list, numElems, 2);

cout << "Your answer is " << value << endl;

return 0;

}

int searchList(int list[], int numElems, int value)

{

int index = 0; // Used as a subscript to search array

int position = -1; // To record position of search value

bool found = false; // Flag to indicate if value was found

while (index < numElems && !found)

{

if (list[index] == value) // If the value is found

{

found = true; // Set the flag

position = index; // Record the value's subscript

}

index++; // Go to the next element

}

return position; // Return the position, or -1

}

// This program is a binary search.

#include <iostream>

using namespace std;

int main()

{

int binarySearch(int array[], int size, int value);

int value;

int array[5] = { 0, 2, 4, 6, 8 };

int size = 5;

value = binarySearch(array, size, 4);

cout << "Your answer is " << value << endl;

return 0;

}

int binarySearch(int array[], int size, int value)

{

int first = 0; // First array element

int last = size - 1; // Last array element

int middle; // Mid point of search

int position = -1; // Position of search value

bool found = false; // Flag

while (!found && first <= last)

{

middle = (first + last) / 2; // Calculate mid point

if (array[middle] == value) // If value is found at mid

{

found = true;

position = middle;

}

else if (array[middle] > value) // If value is in lower half

last = middle - 1;

else

first = middle + 1; // If value is in upper half

}

}

// This program uses the bubble sort function.

#include <iostream>

using namespace std;

int main()

{

const int size = 5; // Size of the array

int array[size] = { 1, 9, 4, 7, 2 }; // The array

void sortArray(int array[], int size); // Function prototype

sortArray(array, size); // Call the sort function

for (int x = 0; x < 5; x++)

{

cout << "The answer is: " << array[x] << endl;

}

return 0;

}

void sortArray(int array[], int size)

{

bool swap;

int temp;

do

{

swap = false;

for (int count = 0; count < (size - 1); count++)

{

if (array[count] > array[count + 1])

{

temp = array[count];

array[count] = array[count + 1];

array[count + 1] = temp;

swap = true;

}

}

}

while (swap);

}

// This program demonstrates a simple class.

#include <iostream>

#include "Header.h"

using namespace std;

// setWidth assigns a value to the width member

void Rectangle::setWidth(double w)

{

width = w;

}

// setLength assigns a value to the length member

void Rectangle::setLength(double len)

{

length = len;

}

// getWidth returns the value in the width member

double Rectangle::getWidth() const

{

return width;

}

// getLength returns the value in the length member

double Rectangle::getLength() const

{

return length;

}

// getArea returns the product of width times length

double Rectangle::getArea() const

{

return width \* length;

}

int main()

{

Rectangle box; // Define an instance of the Rectangle class

double rectWidth; // Local variable for width

double rectLength; // Local variable for length

// Get the rectangle's width and length from the user.

cout << "This program will calculate the area of a\n";

cout << "rectangle. What is the width? ";

cin >> rectWidth;

cout << "What is the length? ";

cin >> rectLength;

// Store the width and the length of the rectangle in the box object

box.setWidth(rectWidth);

box.setLength(rectLength);

// Display the rectangle's data

cout << "Here is the rectangle's data:\n";

cout << "Width: " << box.getWidth() << endl;

cout << "Length: " << box.getLength() << endl;

cout << "Area: " << box.getArea() << endl;

return 0;

}\*/

// This program demonstrates a simple class.

#include <iostream>

#include "Header.h"

using namespace std;

// setAdd assigns a value to the add member

double Calculation::setNum1(double a)

{

num1 = a;

return num1;

}

// setSubtract assigns a value to the sub member

double Calculation::setNum2(double b)

{

num2 = b;

return num2;

}

// getAdd returns the value in the add member

double Calculation::getAdd() const

{

return num1 + num2;

}

// getSubtract returns the value in the sub member

double Calculation::getSubtract() const

{

return num1 - num2;

}

// getMultiply returns the value in the add member

double Calculation::getMultiply() const

{

return num1 \* num2;

}

// getDivide returns the value in the div member

double Calculation::getDivide() const

{

return num1 / num2;

}

int main()

{

Calculation box; // Define an instance of the Rectangle class

double a; // Local variable for add

double b; // Local variable for sub

// Get the rectangle's width and length from the user.

cout << "This program will calculate.\n";

cout << "What is your first number? ";

cin >> a;

cout << "What is your second number? ";

cin >> b;

// Store the add, sub, mult, and div of the calculation in the box object

box.setNum1(a);

box.setNum2(b);

// Display the rectangle's data

cout << "Here is the calculations:\n";

cout << "Add: " << box.getAdd() << endl;

cout << "Subtract: " << box.getSubtract() << endl;

cout << "Multiply: " << box.getMultiply() << endl;

cout << "Divide: " << box.getDivide() << endl;

return 0;

}