Q1)

#include <iostream>

using namespace std;

int add(int num1, int num2) {

return num1 + num2;

}

int main() {

int num1, num2;

cout << "Enter two numbers: ";

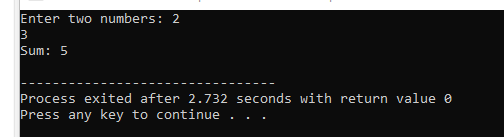
cin >> num1 >> num2;

int sum = add(num1, num2);

cout << "Sum: " << sum << endl;

return 0;

}



Q2)

#include <iostream>

using namespace std;

int main() {

float num1, num2;

cout << "Enter two numbers: ";

cin >> num1 >> num2;

float sum = num1 + num2;

float difference = num1 - num2;

float product = num1 \* num2;

float division = num1 / num2;

cout << "Sum: " << sum << endl;

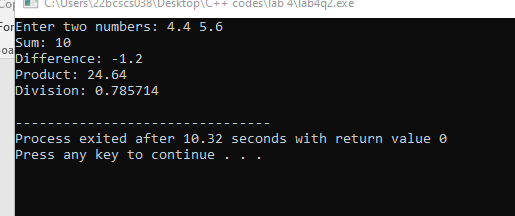
cout << "Difference: " << difference << endl;

cout << "Product: " << product << endl;

cout << "Division: " << division << endl;

return 0;

}



Q3)

#include <iostream>

#include <cmath>

using namespace std;

int main() {

float number;

cout << "Enter a number: ";

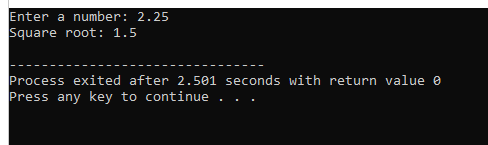
cin >> number;

float squareRoot = sqrt(number);

cout << "Square root: " << squareRoot << endl;

return 0;

}



Q4)

#include <iostream>

#include <algorithm>

using namespace std;

float calculateMean(int array[], int size) {

int sum = 0;

for (int i = 0; i < size; i++) {

sum += array[i];

}

return static\_cast<float>(sum) / size;

}

float calculateMedian(int array[], int size) {

sort(array, array + size);

if (size % 2 == 0) {

return static\_cast<float>(array[size / 2 - 1] + array[size / 2]) / 2;

} else {

return array[size / 2];

}

}

int calculateMode(int array[], int size) {

int mode = array[0];

int maxCount = 1;

int count = 1;

int currentElement = array[0];

for (int i = 1; i < size; i++) {

if (array[i] == currentElement) {

count++;

} else {

if (count > maxCount) {

maxCount = count;

mode = currentElement;

}

count = 1;

currentElement = array[i];

}

}

return mode;

}

int main() {

const int size = 10;

int array[size];

cout << "Enter 10 array elements: ";

for (int i = 0; i < size; i++) {

cin >> array[i];

}

float mean = calculateMean(array, size);

float median = calculateMedian(array, size);

int mode = calculateMode(array, size);

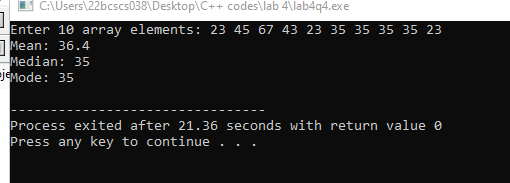
cout << "Mean: " << mean << endl;

cout << "Median: " << median << endl;

cout << "Mode: " << mode << endl;

return 0;

}



Q5)

// I have used string1 as “hello” and string 2 as “world”

#include <iostream>

#include <cstring>

using namespace std;

void concatenateStrings(const char\* str1, const char\* str2) {

char result[100];

strcpy(result, str1);

strcat(result, str2);

cout << "Concatenated string: " << result << endl;

}

int getStringLength(const char\* str) {

return strlen(str);

}

void copyString(const char\* source, char\* destination) {

strcpy(destination, source);

cout << "Copied string: " << destination << endl;

}

void reverseString(const char\* str) {

int length = strlen(str);

char result[100];

for (int i = length - 1, j = 0; i >= 0; i--, j++) {

result[j] = str[i];

}

result[length] = '\0';

cout << "Reversed string: " << result << endl;

}

bool compareStrings(const char\* str1, const char\* str2) {

int result = strcmp(str1, str2);

return result == 0;

}

int main() {

const char\* str1 = "Hello";

const char\* str2 = "World";

concatenateStrings(str1, str2);

int length = getStringLength(str1);

cout << "Length of string 1: " << length << endl;

char destination[100];

copyString(str2, destination);

reverseString(str1);

bool isEqual = compareStrings(str1, str2);

cout << "Are the strings equal? " << (isEqual ? "Yes" : "No") << endl;

return 0;

}

