1.

#include <iostream>

#include <cstdlib>

#include <ctime>

class Student {

private:

char lookup[10]; // private variable

public:

// Constructor

Student() {

// Generate random values for the lookup array

srand(time(NULL));

for (int i = 0; i < 10; i++) {

lookup[i] = 'A' + rand() % 26;

}

}

// Sort the growth signs in one row

void sortGrowthSigns() {

// Sorting algorithm (e.g., bubble sort)

for (int i = 0; i < 9; i++) {

for (int j = 0; j < 9 - i; j++) {

if (lookup[j] > lookup[j + 1]) {

char temp = lookup[j];

lookup[j] = lookup[j + 1];

lookup[j + 1] = temp;

}

}

}

}

// Analyze the private variable to match student marks

void analyzeMarks() {

bool found = false;

for (int i = 0; i < 9; i++) {

if (lookup[i] == lookup[i + 1]) {

found = true;

std::cout << "Found similar sign at index " << i << std::endl;

}

}

if (!found) {

std::cout << "No similar signs found" << std::endl;

}

}

// Print the assessment of information provision in main

void printAssessment() {

std::cout << "Assessment of information provision in Main\_" << std::endl;

// Print the contents of the lookup array

for (int i = 0; i < 10; i++) {

std::cout << lookup[i] << " ";

}

std::cout << std::endl;

}

// Calculate and print the total

void calculateTotal() {

int total = 0;

for (int i = 0; i < 10; i++) {

total += lookup[i];

}

std::cout << "Total: " << total << std::endl;

}

// Calculate and print the average arithmetic third-degree level

void calculateAverage() {

int sum = 0;

for (int i = 0; i < 10; i++) {

sum += lookup[i];

}

double average = static\_cast<double>(sum) / 10;

std::cout << "Average: " << average << std::endl;

}

};

int main() {

Student student;

// Sort growth signs

student.sortGrowthSigns();

// Analyze marks

student.analyzeMarks();

// Print assessment

student.printAssessment();

// Calculate and print total

student.calculateTotal();

// Calculate and print average arithmetic third-degree level

student.calculateAverage();

return 0;

}

2.

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <cstring>

#include <cctype>

void sortArray(int\* arr, int size)

{

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

{

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

{

if (\*(arr + j) < \*(arr + j + 1))

{

int temp = \*(arr + j);

\*(arr + j) = \*(arr + j + 1);

\*(arr + j + 1) = temp;

}

}

}

}

int main()

{

srand(time(NULL));

int arr[10];

std::cout << "Original array: ";

for (int i = 0; i < 10; ++i)

{

arr[i] = rand() % 100; // Filling elements with random numbers between 0 and 99

std::cout << arr[i] << " ";

}

std::cout << std::endl;

sortArray(arr, 10);

std::cout << "Sorted array (descending order): ";

for (int i = 0; i < 10; ++i)

{

std::cout << arr[i] << " ";

}

std::cout << std::endl;

const int balance = 4;

int condition = 3; // Change this value to test different conditions

switch (condition)

{

case 1:

std::cout << "Condition 1: " << balance << std::endl;

break;

case 2:

std::cout << "Condition 2: " << balance << std::endl;

break;

case 3:

std::cout << "Condition 3: " << balance << std::endl;

break;

default:

std::cout << "Default condition: " << balance << std::endl;

break;

}

const char\* str1 = "Hello";

const char\* str2 = "World";

int comparisonResult = strcmp(str1, str2);

if (comparisonResult < 0)

{

std::cout << "String 1 is less than String 2" << std::endl;

}

else if (comparisonResult > 0)

{

std::cout << "String 1 is greater than String 2" << std::endl;

}

else

{

std::cout << "String 1 is equal to String 2" << std::endl;

}

strcat(const\_cast<char\*>(str2), str1);

std::cout << "Concatenated string: " << str2 << std::endl;

int firstCharValue = static\_cast<int>(str1[0]);

std::cout << "Numeric value of the first character of string 1: " << firstCharValue << std::endl;

char str2UpperCase[6];

for (size\_t i = 0; i < 6; ++i)

{

str2UpperCase[i] = std::toupper(str2[i]);

}

std::cout << "String 2 in uppercase: " << str2UpperCase << std::endl;

return 0;

}