Lab task 2

Q.1

#include <iostream>

Using namespace std;

Int main() {

Const int numStudents = 10; // Number of students

Int ages[numStudents]; // Array to store ages

Int maxAge = -1; // Initialize to -1 to handle all positive ages

// Input ages from user

Cout << “Enter the ages of “ << numStudents << “ students:” << endl;

For (int I = 0; I < numStudents; i++) {

Cout << “Student “ << (I + 1) << “: “;

Cin >> ages[i]; // Read age

// Input validation

While (ages[i] < 0) {

Cout << “Please enter a valid age (non-negative): “;

Cin >> ages[i];

}

}

// Find the largest age

For (int I = 0; I < numStudents; i++) {

If (ages[i] > maxAge) {

maxAge = ages[i]; // Update maxAge if current age is greater

}

}

// Output the largest age

Cout << “The largest age among the students is: “ << maxAge << endl;

Return 0;

}

Q2.

#include <iostream>

Using namespace std;

Int main() {

Int size; // Size of the arrays

// Ask user for the size of the arrays

Cout << “Enter the size of the arrays: “;

Cin >> size;

// Dynamically allocate memory for three arrays and one result array

Int\* array1 = new int[size];

Int\* array2 = new int[size];

Int\* array3 = new int[size];

Int\* resultArray = new int[size];

// Input data for the first array

Cout << “Enter elements for Array 1:” << endl;

For (int I = 0; I < size; i++) {

Cout << “Element “ << (I + 1) << “: “;

Cin >> array1[i];

}

// Input data for the second array

Cout << “Enter elements for Array 2:” << endl;

For (int I = 0; I < size; i++) {

Cout << “Element “ << (I + 1) << “: “;

Cin >> array2[i];

}

// Input data for the third array

Cout << “Enter elements for Array 3:” << endl;

For (int I = 0; I < size; i++) {

Cout << “Element “ << (I + 1) << “: “;

Cin >> array3[i];

}

// Add the arrays and store the result in resultArray

For (int I = 0; I < size; i++) {

resultArray[i] = array1[i] + array2[i] + array3[i];

}

// Output the result array

Cout << “The result of adding the three arrays is:” << endl;

For (int I = 0; I < size; i++) {

Cout << “Element “ << (I + 1) << “: “ << resultArray[i] << endl;

}

// Free the dynamically allocated memory

Delete[] array1;

Delete[] array2;

Delete[] array3;

Delete[] resultArray;

Return 0;

}

Q3

#include <iostream>

Using namespace std;

Int main() {

Int size; // Size of the dynamic array

Int target; // The item to search for

Bool found = false; // Flag to check if the item is found

// Ask user for the size of the array

Cout << “Enter the size of the array: “;

Cin >> size;

// Dynamically allocate memory for the array

Int\* array = new int[size];

// Input data for the array

Cout << “Enter “ << size << “ elements:” << endl;

For (int I = 0; I < size; i++) {

Cout << “Element “ << (I + 1) << “: “;

Cin >> array[i];

}

// Ask user for the item to search

Cout << “Enter the item to search for: “;

Cin >> target;

// Perform linear search

For (int I = 0; I < size; i++) {

If (array[i] == target) {

Cout << “Item “ << target << “ found at index “ << I << “.” << endl;

Found = true; // Set flag to true if item is found

Break; // Exit the loop since we found the item

}

}

// If the item was not found, display a message

If (!found) {

Cout << “Item “ << target << “ is not in the list.” << endl;

}

// Free the dynamically allocated memory

Delete[] array;

Return 0;

}