

**ASSIGNMENT NO: Lab 03**

**REGISTRATION NO: 2023-BS-AI-057**

**SUBMITTED BY: Ayesha Imran**

**SUBMITTED TO: Ms. Irsha Qureshi**

**DEPARTMENT: Artificial Intelligence**

**SECTION: A**

**LAB TASK NO 03**

Here are some examples of C++ codes demonstrating the usage of single-dimensional arrays, multi-dimensional arrays, and vectors.

**1. Single-Dimensional Array**

**Example 1:** Initializing and displaying elements of an array.

**Code:**

#include <iostream>

using namespace std;

int main() {

int arr[5] = {10, 20, 30, 40, 50};

for (int i = 0; i < 5; i++) {

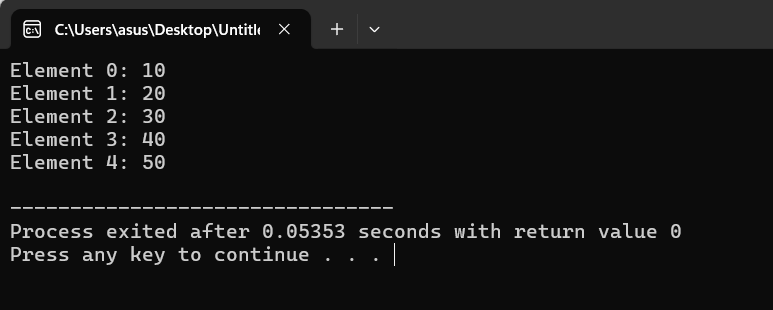
cout << "Element " << i << ": " << arr[i] << endl;

}

return 0;

}

**Output:**



**2. Multi-Dimensional Array**

**Example 2:** Initializing and displaying elements of a 2D array.

**Code:**

#include <iostream>

using namespace std;

int main() {

int arr[2][3] = {{1, 2, 3}, {4, 5, 6}};

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

cout << "Element [" << i << "][" << j << "]: " << arr[i][j] << endl;

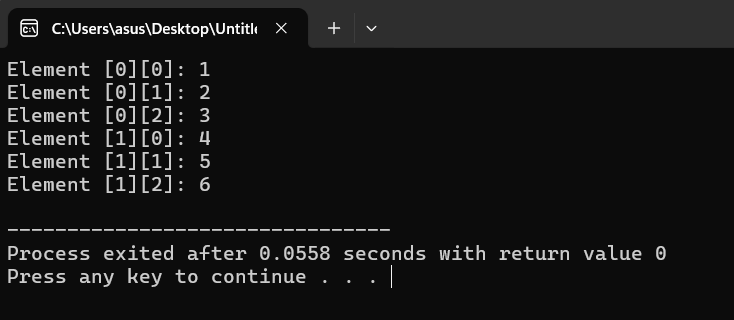
}

}

return 0;

}

**Output:**



**3. Using Vectors**

**Example 3:** Creating a vector and adding elements.

**Code:**

#include <iostream>

#include <vector>

using namespace std;

int main() {

vector<int> vec;

vec.push\_back(10);

vec.push\_back(20);

vec.push\_back(30);

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

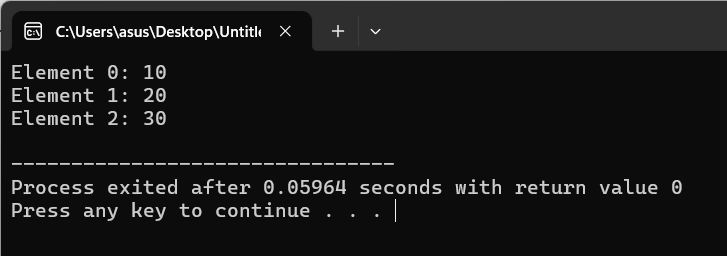
cout << "Element " << i << ": " << vec[i] << endl;

}

return 0;

}

**Output:**



**Additional Examples**

**4. Single-Dimensional Array:** Sum of Elements

**Code:**

#include <iostream>

using namespace std;

int main() {

int arr[5] = {1, 2, 3, 4, 5};

int sum = 0;

for (int i = 0; i < 5; i++) {

sum += arr[i];

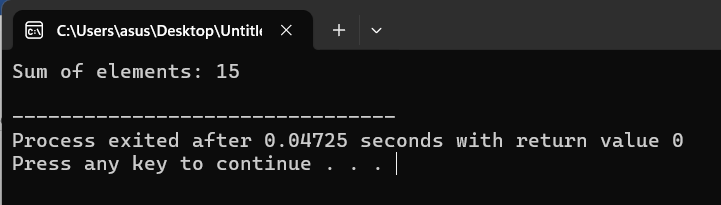
}

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

return 0;

}

**Output:**



**5. Multi-Dimensional Array:** Matrix Addition

**Code:**

#include <iostream>

using namespace std;

int main() {

int A[2][2] = {{1, 2}, {3, 4}};

int B[2][2] = {{5, 6}, {7, 8}};

int C[2][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

C[i][j] = A[i][j] + B[i][j];

cout << "Element [" << i << "][" << j << "]: " << C[i][j] << endl;

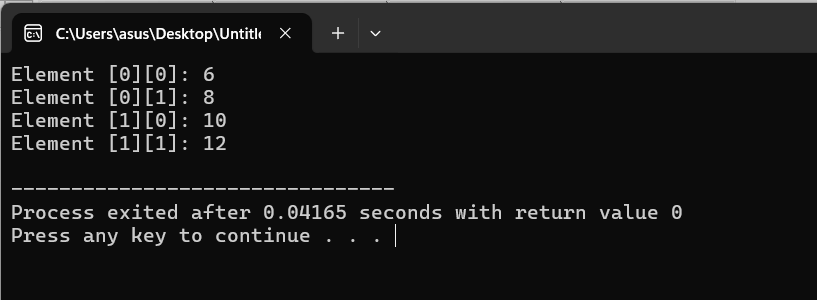
}

}

return 0;

}

**Output:**



**6. Vector:** Removing Elements

**Code:**

#include <iostream>

#include <vector>

using namespace std;

int main() {

vector<int> vec = {10, 20, 30, 40, 50};

vec.pop\_back(); // Remove the last element

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

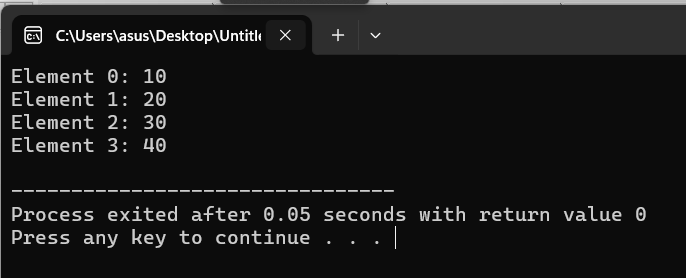
cout << "Element " << i << ": " << vec[i] << endl;

}

return 0;

}

**Output:**



**7. Single-Dimensional Array:** Finding Maximum Element

**Code:**

#include <iostream>

using namespace std;

int main() {

int arr[5] = {5, 3, 8, 6, 2};

int max = arr[0];

for (int i = 1; i < 5; i++) {

if (arr[i] > max) {

max = arr[i];

}

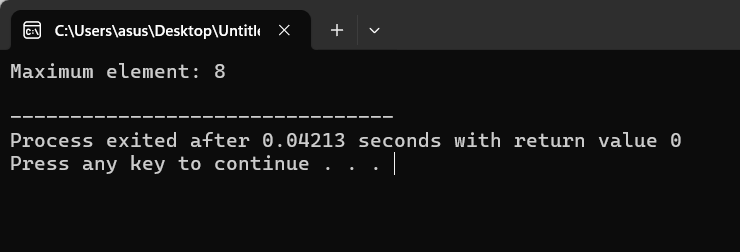
}

cout << "Maximum element: " << max << endl;

return 0;

}

**Output:**



**8. Multi-Dimensional Array:** Transpose of a Matrix

**Code:**

#include <iostream>

using namespace std;

int main() {

int matrix[2][3] = {{1, 2, 3}, {4, 5, 6}};

int transpose[3][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

transpose[j][i] = matrix[i][j];

}

}

// Display the transpose

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 2; j++) {

cout << transpose[i][j] << " ";

}

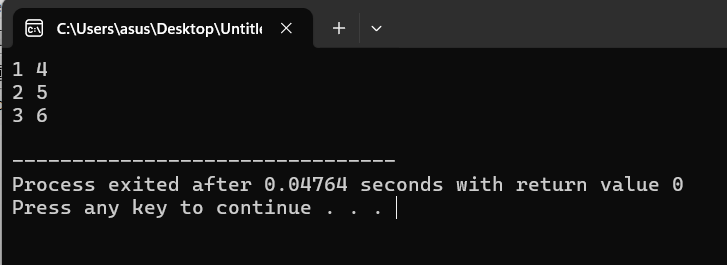
cout << endl;

}

return 0;

}

**Output:**



**9. Vector:** Inserting Elements at Specific Position

**Code:**

#include <iostream>

#include <vector>

using namespace std;

int main() {

vector<int> vec = {1, 2, 4, 5};

vec.insert(vec.begin() + 2, 3); // Insert 3 at the third position

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

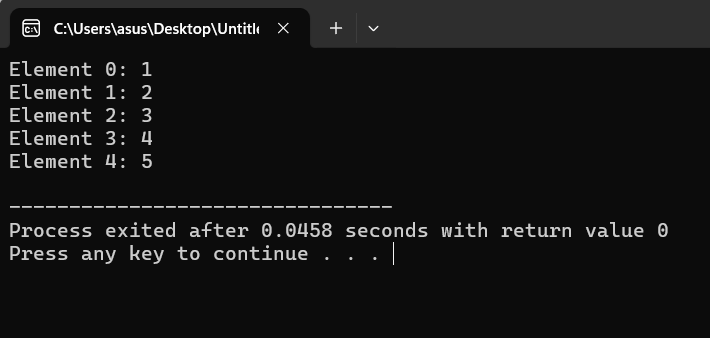
cout << "Element " << i << ": " << vec[i] << endl;

}

return 0;

}

**Output:**



**10. Vector**: Sorting Elements

**Code:**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

vector<int> vec = {5, 2, 8, 1, 3};

sort(vec.begin(), vec.end());

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

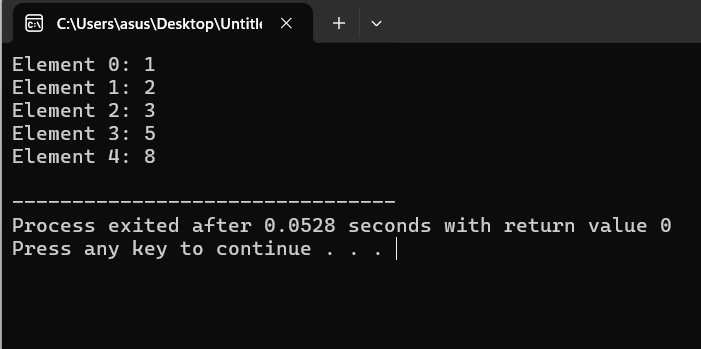
cout << "Element " << i << ": " << vec[i] << endl;

}

return 0;

}

**Output:**



These codes demonstrate how to work with single-dimensional arrays, multi-dimensional arrays, and vectors in C++. Let me know if you need further explanations or more examples!