A close-up of a white background

Description automatically generated

A close up of a computer screen

Description automatically generated

A white rectangular table with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A nested list data structure is essentially a list that contains other lists as its elements. This allows for the creation of multi-dimensional arrays. In C++, this can be represented using vectors of vectors. Here’s an illustration of a nested list:

vector<std::vector<int>> nestedList = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

In this example, nestedList is a vector of vectors, where each element is itself a vector of integers.

Adding new elements to a nested list can affect the structure in several ways:

1. **Adding an Element to an Inner List**: You can add an element to any of the inner lists, which will increase the size of that inner list but not the outer list.
2. nestedList[0].push\_back(4); // Adds 4 to the first inner list
3. **Adding a New Inner List**: You can also add a whole new inner list to the outer list, which increases the size of the outer list.
4. nestedList.push\_back({10, 11, 12}); // Adds a new inner list
5. **Adding Elements with Different Sizes**: When you add new inner lists of different sizes, the nested list becomes jagged, meaning the inner lists do not all have the same size.

nestedList.push\_back({13, 14}); // Adds a new inner list with 2 elements