 Match each of the vocabulary words at the right with the BEST definition on the left.

* A contiguous block of memory used to hold a set of data of the same type →
  + Array,
* These characters are used when defining an array to enclose its intended size. →
  + [ ],
* A named memory location used to store data that does not change during program execution →
  + constant,
* This looping construct is optimal for working with arrays →
  + for,
* An example of a simple selection statement →
  + if,
* Can be used to store a set of data that is "tabular" in nature →
  + 2 D array,
* Starting Index in an Array →
  + Zero,
* You use this search when looking for something in an ordered array →
  + binary search,
* This sort is good when adding items to a nearly sorted array →
  + Insertion,
* This sort is good when you are looking for the three smallest items, for example →
  + Selection

The Answer search repeatedly divides portion of the array being searched in half when it does not find a value.

* The maximum number of comparisons performed by linear search to find an item in an array of N elements is:
  + N
* How many numbers are placed in order on each pass through the data (each iteration of the inner loop) for the selection sort? (Enter a numeral.)
  + 1
* When storing values in an array, C++ does not allow you to access locations outside the array.
  + False

#include <iostream>  
#include <iomanip>

**using** **namespace** std;

**const** **int** SIZE = 5;

// fill table with multiplication table 1-5  
**void** filler(**int** table[][SIZE])  
{  
    **for** (**int** i = 0; i < SIZE; i++)  
**for**(**int** j = 0; j < SIZE; j++)  
           table[i][j] = (i+1)\*(j+1);  
}

// display table, one row per line  
**void** display(**int** table[][SIZE])  
{  
**for** (**int** i = 0; i < SIZE; i++)  
     {  
**for**(**int** j = 0; j < SIZE; j++)  
        {  
            cout << setw(5) << table[i][j];  
        }  
        cout << endl;  
    }  
        cout << endl;  
}

// define table, fill it, display it  
**int** main()  
{  
**int** table[SIZE][SIZE];

    filler(table);

    display(table);

**return** 0;  
}