```cpp
// This program will read in prices and store them into a two-dimensional
 array.
// It will print those prices in a table form.

// Connor Seemann

//preprocessors to include for the program
#include <iostream>
#include <iomanip>
#include <string>

// prototypes for functions
void swap(char &a, char &b);
void BubbleSort(char array[], int size);
void makeFrequency(char grades [], int size, int frequency [], int range, int
 MAX_SIZE);

int main()
{
    int max;
    do
    {
        std::cout << "Please enter the amount of grades to be read in (1-50)"
         << std::endl;
        std::cin >> max;
        if (max <= 50 && max >= 1)
            break;
        else
            std::cout << "Please enter numbers in the correct range of 1-50"
             << std::endl << std::endl;

    } while (!(max <= 50 && max >= 1));

    // Define constant variables
    const int MAX_SIZE = max;       // This will set the constant MAX_RANGE to
     temp
    const int MAX_RANGE = max;      // This will set the constant MAX_RANGE to
     temp
    typedef char GradeList[MAX_SIZE];// defining the GradeList typedef for
     cleaner code

    GradeList ListOfGrades;         // declaring list of grades
    char temp;                       // declaring temp
    int size = 0;                   // declaring size
    int frequency[MAX_RANGE];       // declaring frequency

    // start main program
    // getting input

    std::cout << "All grades must be upper case A B C D or F" << std::endl;
    do
```

```
    {
        // breaks if the grade entered is -99
        std::cout << "Please enter a grade:" << std::endl;
        std::cin >> temp;   // getting the input
        if (temp == 'A' || temp == 'B' || temp == 'C' || temp == 'D' || temp
         == 'F')
        {
            ListOfGrades[size] = temp;
            size++;                          // incrementing size
        }

        else
            std::cout << "Thant is not an accepted grade... \
            \nplease make sure the grades you enter are A B C D or F\n" <<
             std::endl;

    } while (size < MAX_SIZE);

    // start function calls

    BubbleSort(ListOfGrades, size);
        // sorts the array in order least to greatest

    makeFrequency(ListOfGrades, size, frequency, MAX_RANGE, MAX_SIZE);
        // call for makeFrequency

return 0;
}

//*************************************************************************
 ***
// swap
//
// task:          This function will take two inputs and with those two inputs
//                it will swap them.
// data in:       two integers to be swapped.
// data returned: two integers swapped.
//
//*************************************************************************
 ***

void swap(char &a, char &b)
{
    int temp = b;
    b = a;
    a = temp;

    return;
}
```

```
//*************************************************************************
 ***
// BubbleSort
//
// task:          This function will sort an input array to be sorted in
 numerical
//                order (least to greatest).
// data in:       array of integers and the size in an integer.
// data returned: array in numerical order.
//
//*************************************************************************
 ***

void BubbleSort(char array[], int size)
{
    for (int maxElement = size - 1; maxElement > 0; --maxElement)   // for
     loop to iterate through the array the size - 1 times
    {
        for (int index = 0; index < maxElement; ++index)            // for
         loop to iterate throught the array
        {
            if (array[index] > array[index + 1])                // checks to
             see if the next index is grater than the current
            {
                swap(array[index], array[index + 1]);           // calls the
                 swap function to switch the two values
            }
        }
    }
}

//*************************************************************************
 ***
// makeFrequency
//
// task:          In this function the program will compair numbers in the
 grades[]
//                array and will add the frequency in the frequency[] array
 after
//                it will output the frequency in a clean format.
// data in:       Array of number and the size of that array, frequency array
 and
//                the possible range of the arrays.
// data returned: returns the data in the grades[] array and returns the
 frequency
//                of the data in the grades[] array.
//
//*************************************************************************
 ***
```

```
void makeFrequency(char grades [],      int size,          // getting the
 input of grades and size
                   int frequency[5], int range,
                   int MAX_SIZE)         // getting the input of frequency and
                    range
{
    std::cout << std::endl << "----------- FINDING THE FREQUENCY OF NUMBERS
     -----------" << std::endl << std::endl;

    for (int i = 0; i < range; i++)                      // creating a blank
     array that is in the same range
        frequency [i] = 0;                               // setting the array
         of frequency to zeros

    for (int i = 0; i < size; i++)
        std::cout << grades[i] << ' ';                     // print statement
         that will print the numbers in the array
        // std::cout << std::endl;                          // neatens up the
         output if the testing print for frequency is enabled

    int temp = 0;
    std::cout << std::endl;
    for (int i = 0; i < size; i++)                       // scanning numbers
     and building the frequency array
    {
        // std::cout << frequency[grades[i]] << " ";       // testing print
         statement for the frequency of numbers
        if (grades[i] == 'A')
            temp = 0;
        else if (grades[i] == 'B')
            temp = 1;
        else if (grades[i] == 'C')
            temp = 2;
        else if (grades[i] == 'D')
            temp = 3;
        else if (grades[i] == 'F')
            temp = 4;
        std::cout << temp << " ";
        frequency[temp]++;
    }
    std::cout << std::endl << std::endl;

    std::cout << "Frequency of ages are:" << std::endl;
    for (int i = 0; i < size; i++)                       // The following loop
     outputs the frequencys for the ages entered
    {
        for (int x = 1; x < MAX_SIZE; x++)               // gets the integer
         for the grades array to be checked and adds x   |
        {
```

```
            //std::cout << grades[i] << " " << grades[i + 1] << " " << i <<
             std::endl;                                          // and when it
             breaks thats the amount of that number is how many |
            if (grades[i] != grades[i+x])
            {

                if (grades[i] == 'A')
                    temp = 0;
                else if (grades[i] == 'B')
                    temp = 1;
                else if (grades[i] == 'C')
                    temp = 2;
                else if (grades[i] == 'D')
                    temp = 3;
                else if (grades[i] == 'F')
                    temp = 4;
                std::cout << std::setw(5) << grades[i] << std::setw(5) << " :
                 " << std::setw(5) << frequency[temp] << std::endl;

            }
            break; // break out of x incrementing i
        } // for x

    } // for i
    std::cout << std::endl;

return;
}
// END OF PROGRAM
```