Possible Quiz Questions

1) Write a function called \textit{gapSum} that calculates the sum of the gaps between adjacent entries of an array. (A gap between two numbers is the absolute value of their difference.) For example, a program that uses the function \textit{gapSum} as follows.

```c
int main() {
    int x[5] = {3, 1, 4, 1, 5};
    cout << gapSum(x, 5) << endl; // prints 12
    // The gaps are 2, 3, 3, 4 and these add to 12
    return 0;
}
```

2) Write a function called \textit{maxGap} that calculates the biggest gap between adjacent entries of an array. (A gap between two numbers is the absolute value of their difference.) For example, a program that uses the function \textit{maxGap} follows.

```c
int main() {
    int x[5] = {3, 1, 4, 1, 5};
    cout << maxGap(x, 5) << endl;
    // prints 4 corresponding to the gap from 1 to 5.
    return 0;
}
```

3) Write a C++ function called \textit{squares} that replaces each element of a 2-dimensional array (with two columns) by its square. It should be possible to use your function in the following program.

```c
int main() {
    int data[2][2] = {{1, 2}, {3, 4}};
    squares(data, 2, 2);
    for (int i = 0; i < 2; i++)
        cout << data[1][i] << " "; // prints 9 16
}
```

4) Write a function called \textit{randSelect} that selects one column at random in a 2-dimensional array of integers and returns the product of the entries in that column. (You must use an appropriate standard C++ function to generate all random numbers.) For example, a program that uses the function \textit{randSelect} follows.

```c
int main() {
    int x[2][3] = {{3, 1, 4}, {1, 5, 9}};
    cout << randSelect(x, 2, 3);
    // might print 36 if the last col is selected
    cout << endl;
    return 0;
}
```
5) Write a function called `numPositive` that finds the number of rows with positive sum in a 2-dimensional array of decimals that has 4 columns. The array and the capacities are parameters. (Note that 0 is not positive.) For example, a program that uses the function follows.

```cpp
int main() {
    double d[2][4] = {{2, 4, -6, -8}, {-1, -3, 5, 1.5}};
    cout << numPositive(d, 2, 4) << endl;
    // prints 1 because only one row, the 2nd has a positive sum
    return 0;
}
```