2.1

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/vector_squared
Original vector: 1.50 2.70 3.20 4.90 5.10
Squared vector: 2.25 7.29 10.24 24.01 26.01
Distance from origin (using accumulate): 8.35
Distance from origin (using inner_product): 8.35
Distance from origin (using accumulate with lambda): 8.35
```

2.2

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/median
Testing with odd number of elements: 3.1 1.4 7.2 4.9 2.8
Median (sort): 3.1
Median (partial_sort): 3.1
Median (nth_element): 3.1
Median (template): 3.1

Testing with even number of elements: 3.1 1.4 7.2 4.7 2.8 5.5
Median (sort): 3.9
Median (partial_sort): 3.9
Median (nth_element): 3.9
Median (template): 3.9
```

sort:

Uses full sorting (O(n log n)) and is the least efficient but most straightforward approach.

partial_sort:

More efficient than full sort because it only sorts elements up to the median position. It's still O(n log n) in the worst case, but with a smaller constant factor since it doesn't fully sort the latter half of the vector.

nth_element:

Most efficient of the three, with O(n) average time complexity. It's faster because it only ensures that elements before the nth position are smaller than the nth element and elements after the nth position are larger than the nth element. It doesn't sort any portions of the array

2.3

```
weicaivi@linux6:~/mpcs51044-cpp/build$ /home/weicaivi/mpcs51044-cpp/build/pascals_triangle
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5  10  10   5   1
    1   6  15  20  15   6   1
  1   7  21  35  35  21   7   1
```

The class-oriented version makes the code better:

- The code is more structured - data and related functions are grouped together in a logical way.
- Once you create a triangle object, you can easily reuse it, access its values, or print it multiple times without recalculating.

- Data is protected (private) and can only be accessed through controlled methods, preventing accidental modifications.

Trade-off: It's slightly more complex than the original procedural version, so for a very simple one-time use, the original might be fine.