```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <omp.h>

using namespace std;

const int VECTOR_SIZE = 100;

int main()
{
    // initialize random seed
    srand(time(NULL));

    // allocate memory for the vectors
    int* vector1 = new int[VECTOR_SIZE];
    int* vector2 = new int[VECTOR_SIZE];
    int* result = new int[VECTOR_SIZE];

    // fill the vectors with random numbers
    #pragma omp parallel for
    for (int i = 0; i < VECTOR_SIZE; i++)
    {
        vector1[i] = rand() % 10000;
        vector2[i] = rand() % 10000;
    }

    // add the vectors in parallel using OpenMP
    #pragma omp parallel for
    for (int i = 0; i < VECTOR_SIZE; i++)
    {
        result[i] = vector1[i] + vector2[i];
    }

    // print the first and second vectors and their sum
    cout << "Vector 1: \n[";
    for (int i = 0; i < VECTOR_SIZE; i++)
    {
        cout << vector1[i];
        if (i != VECTOR_SIZE - 1)
        {
            cout << ", ";
        }
    }
    cout << "]" << endl;

    cout << "Vector 2:\n [";
    for (int i = 0; i < VECTOR_SIZE; i++)
    {
        cout << vector2[i];
        if (i != VECTOR_SIZE - 1)
        {
            cout << ", ";
        }
    }
    cout << "]" << endl;

    cout << "Result: \n[";
    for (int i = 0; i < VECTOR_SIZE; i++)
    {
        cout << result[i];
        if (i != VECTOR_SIZE - 1)
        {
            cout << ", ";
        }
    }
    cout << "]" << endl;

    // free the allocated memory
    delete[] vector1;
    delete[] vector2;
    delete[] result;

    return 0;
}
```