Write a C++ program that finds and reports the squared length of the vector that is the result of the cross product of two, 3-dimensional vectors. Got that? If not, don't worry. Your teacher has provided a C++ function that works.

More specifically, your `main` function should first ask for and read in two different x values, two different y values, and two different z values. Your `main` function should then call the function named `SqrLenCrossProd()`, the body of which you should write using Microsoft's ***in-line*** assembly code. You may assume that the result of all multiplication, addition and subtraction operations will not exceed the limits of a 32-bit signed integer.

To help with the mathematical formulas, your professor has written a C++ version of the `SqrLenCrossProd()` function, as seen below. Your program should replace the statements printed in **bold** with in-line assembly instructions.

```
int SqrLenCrossProd(     int x1,  int y1,  int z1,   // Input vector 1
                         int x2,  int y2,  int z2)   // Input vector 2
{
      int x3, y3, z3;
      int answer;

      x3 = y1*z2 – y2*z1;
      y3 = x2*z1 – x1*z2;
      z3 = x1*y2 – x2*y1;

      answer = (x3*x3 + y3*y3 + z3*z3);

      return answer;
}
```

← **Replace these statements with assembly code**

Be sure to include comments for your assembly code using the C++ code as the comments.

In order to test your code you will need to write a `main()` function that calls the `SqrLenCrossProd()`. You're free to design the `main()` function anyway you like – as long as it passes 6 integers to the function and then displays the returned integer value. However, you will not submit your `main()` function. The professor will use his own version.

Sample Runs:

```
Enter vector #1: 1 2 3
Enter vector #2: 1 1 1
Answer = 6

Enter vector #1: 1 0 0
Enter vector #2: 0 1 0
Answer = 1

Enter vector #1: 10 10 10
Enter vector #2: -10 50 -15
Answer = 785000
```