

Problem #1
Vector

A vector in 3d space can be represented by an ordered group of 3 real numbers. These 3 numbers represent its strength along x , y and z axes respectively. For example, the vector (2, 3, -5) has 2 units strength along x -axis, 3 units along y -axis and 5 units along the z -axis in negative direction. Following are some operations that can be performed on vectors:

Let U represents the vector (u_1, u_2, u_3) , and V represents the vector (v_1, v_2, v_3) . Then,

- The result of adding vector U and vector V is a vector $(u_1 + v_1, u_2 + v_2, u_3 + v_3)$
- Subtracting vector V from vector U results in a vector $(u_1 - v_1, u_2 - v_2, u_3 - v_3)$
- Dot product of vectors U and V is the real number $u_1.v_1 + u_2.v_2 + u_3.v_3$
- Cross product of vectors U and V is a vector $(u_2.v_3 - u_3.v_2, u_3.v_1 - u_1.v_3, u_1.v_2 - u_2.v_1)$
- The magnitude of vector U is the real number $\sqrt{u_1^2 + u_2^2 + u_3^2}$

You need to implement a class called **Vector** that supports the above mentioned functionalities. Store the strength of the vector along x , y and z axis in 3 private double variables called x , y and z respectively. Instances of Vector class should be initialized to (0,0,0). Write appropriate constructor and implement necessary methods so that the following code (in the next page) in the main function works perfectly with your class. (Carefully review the code and comments therein to find out what needs to be implemented in your class.)

Hint: From the code of main(), first identify what constructors and functions should be part of Vector class. Then identify what should their prototype look like. Then write the prototypes along with empty body, and ensuring a return statement that is in accordance with the return type mentioned in the prototype. After you have done this, your code should start compiling. Then write the necessary logic/calculations in each method.

```
int main()
{
    int i;
    char cmd[20];
    Vector vector[2], result, zero;
    double a, b, c;

    while (EOF != scanf("%s", cmd))
    {
        for (i = 0; i < 2; i++)
        {
            scanf("%lf%lf%lf", &a, &b, &c);

            // Sets the strength of vector[i] along x, y and z axes
            // to a, b, c respectively.
            vector[i].Set(a, b, c);

            if (vector[i].Equals(zero))
            {
                cout << "You entered a zero vector" << endl;
            }
            else
            {
                // prints the magnitude of vector[i]
                cout << "You entered a vector of "
                     << vector[i].Magnitude() << " magnitude" << endl;
            }
        }

        cout << "Result of " << cmd << " operation is: ";

        if (0 == strcmp(cmd, "dot"))
        {
            cout << vector[0].Dot(vector[1]) << endl;
            continue;
        }

        if (0 == strcmp(cmd, "add"))
        {
            result = vector[0].Add(vector[1]);
        }
        else if (0 == strcmp(cmd, "subtract"))
        {
            result = vector[0].Subtract(vector[1]);
        }
        else if (0 == strcmp(cmd, "cross"))
        {
            result = vector[0].Cross(vector[1]);
        }

        // Prints the vector as "(u1, u2, u3)", where u1, u2, u3
        // are respectively the strength of the vector along x, y, z axes
        result.Print();
    }
    return 0;
}
```