**CodinGame Puzzle Solutions**

**The Descent**

int main()

{

    // game loop

    while (1) {

        int mountain\_highest;

        mountain\_highest = 0;

        int mountain\_to\_shoot;

        mountain\_to\_shoot = 0;

        for (int i = 0; i < 8; i++) {

            int mountain\_h; // represents the height of one mountain.

            cin >> mountain\_h; cin.ignore();

            if (mountain\_highest < mountain\_h)

            {

                mountain\_highest = mountain\_h;

                mountain\_to\_shoot = i;

            }

        }

        // Write an action using cout. DON'T FORGET THE "<< endl"

        // To debug: cerr << "Debug messages..." << endl;

        cout << mountain\_to\_shoot << endl; // The index of the mountain to fire on.

    }

}

**The Power of Thor**

int main()

{

    int light\_x; // the X position of the light of power

    int light\_y; // the Y position of the light of power

    int initial\_tx; // Thor's starting X position

    int initial\_ty; // Thor's starting Y position

    cin >> light\_x >> light\_y >> initial\_tx >> initial\_ty; cin.ignore();

    int current\_tx;

    current\_tx = initial\_tx;

    int current\_ty;

    current\_ty = initial\_ty;

    string dir;

    // game loop

    while (1) {

        int remaining\_turns; // The remaining amount of turns Thor can move. Do not remove this line.

        cin >> remaining\_turns; cin.ignore();

        if      ((current\_tx > light\_x) && (current\_ty > light\_y)) {current\_tx -= 1; current\_ty -= 1; dir = "NW";}

        else if ((current\_tx < light\_x) && (current\_ty > light\_y)) {current\_tx += 1; current\_ty -= 1; dir = "NE";}

        else if ((current\_tx < light\_x) && (current\_ty < light\_y)) {current\_tx += 1; current\_ty += 1; dir = "SE";}

        else if ((current\_tx > light\_x) && (current\_ty < light\_y)) {current\_tx -= 1; current\_ty += 1; dir = "SW";}

        else if (current\_tx > light\_x) {current\_tx -= 1; dir = "W";}

        else if (current\_tx < light\_x) {current\_tx += 1; dir = "E";}

        else if (current\_ty < light\_y) {current\_ty += 1; dir = "S";}

        else if (current\_ty > light\_y) {current\_ty -= 1; dir = "N";}

        // Write an action using cout. DON'T FORGET THE "<< endl"

        // To debug: cerr << "Debug messages..." << endl;

        // A single line providing the move to be made: N NE E SE S SW W or NW

        cout << dir << endl;

    }

}

**UNARY CODE – NOT FINISHED**

int main()

{

    string message;

    getline(cin, message);

    string res = " ";

    int pre = -1;

    for (size\_t i = 0; i < message.**length**(); ++i)

    {

        for (size\_t j = 6; j >= 0; --j)

        {

            size\_t bit = message.at(i) >> j & 1;

            {

            if (bit != pre)

            {

                if (-1 != pre)

                {

                    res += " ";

                }

                res += 1 == bit ? "0 " : "00 ";

                pre = bit;

            }

            res += "0";

            }

        }

    }