

## Sąlygos sakiniai

### 1 užduotis. Onboarding <https://www.codingame.com/training/easy/onboarding>

Reikia papildyti programą sakiniais, kurie leistų sunaikintų artimiausio priešo laivą. enemy1 yra pirmo priešo vardas, dist1 – atstumas iki enemy1 priešo, enemy2 – antro priešo vardas, dist2 – atstumas iki enemy2 priešo.

```
#include <iostream>

using namespace std;

int main()
{
    // game loop
    while (1) {
        string enemy1; // 1 priešo vardas
        cin >> enemy1; cin.ignore();
        int dist1; // atstumas iki pirmo priešo
        cin >> dist1; cin.ignore();
        string enemy2; // 2 priešo vardas
        cin >> enemy2; cin.ignore();
        int dist2; // atstumas iki 2 priešo
        cin >> dist2; cin.ignore();

        // Čia turi būti įrašytas programos kodas, kuris suranda, iki kurio
        // priešo yra mažesnis atstumas

        if (dist1 < dist2) {
            cout << enemy1 << endl;
        } else {
            cout << enemy2 << endl;
        }
    }
}
```

## 2 užduotis. Thoro galia (Power of Thor). <https://www.codingame.com/training/easy/the-descent>

Thoras, eidamas iš nurodyto taško, turi pasiekti šilumos šaltinį.

Thoras juda stačiakampyje, kurio plotis yra 40, o ilgis 18 vienetų. Stačiakampio koordinatų pradžia – viršutinis kairysis kampas. Kairiojo viršutinio kampo koordinatės yra (0; 0), dešiniojo apatinio – (39, 17).

Pradiniai duomenys: šilumos šaltinio koordinatės (lightX; lightY), Thoro buvimo vietos koordinatės (initialTX; initialTY). Programa turi išvesti, kuria kryptimi turi judėti Thoras (N, NE, E, SE, S, SW, W, NW).



N (North)  
NE (North-East)  
E (East)  
SE (South-East)  
S (South)  
SW (South-West)  
W (West)  
NW (North-West)

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    int lightX; // Šviesos šaltinio X koordinatė
    int lightY; // Šviesos šaltinio Y koordinatė
    int initialTX; // Thoro buvimo vietos X koordinatė
    int initialTY; // Thoro buvimo vietos Y koordinatė
    cin >> lightX >> lightY >> initialTX >> initialTY; cin.ignore();

    while (1) {
        int remainingTurns; // Kiek posūkių Thoras gali padaryti
        cin >> remainingTurns; cin.ignore();

        // Turi būti išvedama: N, NE, E, SE, S, SW, W arba NW
        cout << "?" << endl;
    }
}
```

## 3 užduotis. Marsaeigis (Mars Lander) <https://www.codingame.com/training/easy/mars-lander-episode-1>

**5 uždutis. Saugiklis (Blowing fuse)** <https://www.codingame.com/training/easy/blowing-fuse>

**6 uždutis. Strėlė (The dart 101)** <https://www.codingame.com/training/easy/the-dart-101>

### Ciklai, sąlygos sakiniai

**1 uždutis. Nusileidimas (The descent).** <https://www.codingame.com/training/easy/the-descent>

Parašyti programos kodą, kuris leistų rasti aukščiausio kalno numerį. Iš viso yra 8 kalnai, kalnų numeracija pradedama nuo nulio.

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    // game loop
    while (1) {
        for (int i = 0; i < 8; i++) {

            int mountainH; // vieno kalno aukštis
            cin >> mountainH; cin.ignore();

        }

        cout << ? << endl; // išvedamas kalno numeris
    }
}
```

**2 uždutis. Žirgų lenktynės (Horse-racing duals)**  
<https://www.codingame.com/training/easy/horse-racing-duals>